Computers Math. Applic. Vol. 23, No. 2–5, pp. 1–50, 1992
Printed in Great Britain.

0097-4943/92 $5.00 + 0.00

SEMANTIC NETWORKS

FRITZ LEHMANN
124 Parsons Avenue, Webster Groves, Missouri 63119, U.S.A.

Abstract — A semantic network is a graph of the structure of meaning. This article introduces semantic network systems and their importance in Artificial Intelligence, followed by I. the early background; II. a summary of the basic ideas and issues including link types, frame systems, case relations, link valence, abstraction, inheritance hierarchies and logic extensions; and III. a survey of 'world-structuring' systems including ontologies, causal link models, continuous models, relevance, formal dictionaries, semantic primitives and intersecting inference hierarchies. Speed and practical implementation are briefly discussed. The conclusion argues for a synthesis of relational graph theory, graph-grammar theory and order theory based on semantic primitives and multiple intersecting inference hierarchies.

... when controversies arise, it will not be a work of learned disputation between two philosophers, but between two computists. It will be enough for them to take pen in hand, sit at the abacus, and say to each other, as friends: 'Let us calculate!'
— Leibniz

All thought is diagrammatic.
— Charles S. Peirce

1. INTRODUCTION

A computer or robot seems stupid when you have to tell it exactly what to do and how to do it. One aim of Artificial Intelligence (AI) is to let you just describe your problem and have the machine solve it with general reasoning techniques. Typically, a general-purpose reasoning program operates on a formal description of the particular problem. Like a capable human being, the program may need to use background knowledge of the subject area along with general common sense knowledge about the world. Somehow this knowledge must be represented in the machine. In the last 15 years attention in AI has shifted away from reasoning programs to knowledge representation as the primary challenge. Instead of using natural languages (which are highly arbitrary and ambiguous), such knowledge is often represented using abstract conceptual structures called semantic networks.

Certain computing tasks vital to industry, the professions, and the military have reached a practical limit beyond which conventional computing (ordinary data processing and mathematical modelling) cannot go. These tasks require explicit, in-depth *conceptual analysis*, rather than just repetitive processing of the elements of a model. In an AI system, the concepts and principles of the subject domain are arranged in an ordered structure called a *Knowledge Base*. Transcending mere storage and retrieval of asserted facts, the computer uses this structure to infer other knowledge from that which has been stored directly. This depends on using the fundamental *semantic* structure of the concepts involved, as opposed to the *syntactic* (grammar) structure of any particular language.

Several quite different sentences in English (or other languages) can all have the same essential meaning and underlying semantic structure: a network of interrelated conceptual units. See Figure 1. This is a 'map' of the meaning. A network is also a convenient way to organize information in a computer or database.

Toby the hungry tiger follows his mother.
Toby, the son of the tigress he follows, is hungry.
The tiger is followed by her hungry cub Toby.
Hunger grips Toby, son of the tigress leading him.
She who bore hungry tiger Toby
is also by him followed.
Der Tiger Toby, der seiner
Mutter folgt, hat Hunger.
Hungry Toby's relation to the tiger
is one of mother-following.
$\exists$(Toby)$\exists(x)$(Tiger(Toby) $\wedge$
Tiger(x) $\wedge$ Follows(Toby,x) $\wedge$
Mother-of(x,Toby) $\wedge$ Female(x) $\wedge$
Male(Toby) $\wedge$ Hungry(Toby)).

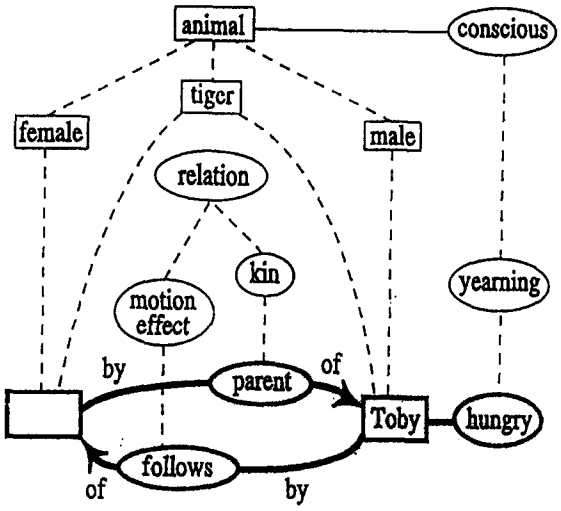


Figure 1. All of these sentences share an underlying meaning structure. A semantic network is on the right. The heavy lines are asserted relation-links between described individuals, forming a *relational graph*. The dashed lines are IS-A links in an *abstraction hierarchy* of categories used to infer features of individuals, qualities and relations. (Here 'follows' and 'parent' may be considered as relations, or as conceptual units with 'by' and 'of' as their relations.)

A **semantic network** or **net** represents knowledge as a net-like graph.[1] An idea, event, situation or object almost always has a composite *structure*; this is represented in a semantic network by a corresponding structure of **nodes** (drawn as circles or boxes) representing *conceptual units*, and directed **links** (drawn as arrows between the nodes) representing the *relations* between the units. The network in Figure 1 has a *relational graph* describing two individuals (Toby and the unnamed tigress) with their asserted qualities and relations, on top of which is an *abstraction hierarchy* of more general concepts and relations. From this combined structure it is possible to deduce things about the composite concept as a whole and its relations to other concepts.

An abstract (graph-theoretic) network can be diagrammed, defined mathematically, programmed in a computer, or hard-wired electronically. It becomes *semantic* when you assign a meaning to each node and link. Unlike specialized networks and diagrams, semantic networks aim to represent any kind of knowledge which can be described in natural language. A semantic network system includes not only the explicitly stored net structure but also methods for automatically deriving from that a much larger structure or body of *implied* knowledge. For example, the assertion in Figure 1 that Toby is hungry implies that he is a conscious animal, and everything true of conscious animals is automatically true of Toby. Almost all systems have structured concept-hierarchies or taxonomies used for this kind of derivation (described in Section 6), and these hierarchies themselves are also 'semantic networks.'

In the 1970's, semantic network research emphasizing this 'structure of knowledge' approach became predominant in AI, later contending with *Rule-based Expert Systems* for center stage. Since then it has waxed and waned periodically and many of its ideas reappear in other guises such as *Object-Oriented Systems*. Many Expert Systems now include Object-Oriented extensions which allow easy implementation of semantic networks (see Section 15). The latest vogue in AI, *neural nets*, often seems to be an opposing, anti-analytic approach using no identifiable symbols or concepts in the computer at all, but there are systems (Section 14) which have enough internal compositional structure to be used 'semantically.' Semantic networks are seldom claimed to exist physically in the brain; rather, they are viewed as idealized reasoning structures with practical

[1] Throughout this volume, "graph" means an interconnected vertex-and-arc (dot-and-line) structure as studied in Graph Theory rather than a graph plotted in Cartesian X-Y coordinates.

computer application. A common goal of this kind of AI is to impart obvious 'common sense' to computers.[2]

The specialized semantic network inference methods discussed below are often combined with other standard AI reasoning methods such as rule-based search, automatic theorem-proving, constraint satisfaction, machine learning algorithms, and others.

Semantic networks are used in almost every application area of AI, including natural language understanding, deductive databases, library document retrieval, business planning, medical diagnosis, legal case analysis, analogical reasoning, expert systems, robot control, intelligent Computer Aided Design, visual pattern recognition, simulated aircraft control, and many more.

1.1. Overview

There are now eight major research families of semantic network systems plus countless independent projects around the world. The eight are:

CONCEPTUAL DEPENDENCY • CONCEPTUAL GRAPHS • ECO • KL-ONE
PATH-BASED INHERITANCE • PREFERENCE SEMANTICS • PSN • SNePS

In some of these there are all sorts of variants and it is quite confusing to the newcomer. The eight invited survey articles following this one introduce the basic ideas and give some guidance to the research directions within each family. These are followed by 25 articles on diverse subjects in the field.[3] The surveys and articles cover most contemporary work. In this article, following this introduction, **Part I** describes the historic origins of semantic networks; **Part II** covers the basics such as frame systems, relational graphs, deep case relations, link valence, inheritance hierarchies, IS-A links, relational inheritance and logic extensions; and **Part III** discusses 'world-structures': ontologies, continuous models, relevance, dictionaries, semantic primitives and intersecting inference hierarchies. I then briefly treat some speed and implementation issues. (If you're very familiar with semantic networks you can skip **Part II**.) My emphasis will be on knowledge structures rather than connections with natural language or particular notations, and I'll mention how the other articles relate to the subject.

1.2. Flat or Deep?

Most commercial AI systems are Rule-based Expert Systems with large sets of IF-THEN rules supplied by an expert in some field to embody his or her expertise. There is near-universal disappointment at the 'flatness,' or lack of structured knowledge, in Expert Systems based on rules whose symbols may represent anything at all. It is hard just to keep track of numerous *arbitrarily* interrelated rules, and, during automatic inference, the computational burden of exhaustive search through a large space of unconstrained rules is often overwhelming. Serious users are soon frustrated by the indiscriminate stupidity of pure rule-based systems. A knowledgeable person will spot obvious errors in a description whereas a dolt will just accept it. A typical Expert System often seems clever due to the suggestive names of internally meaningless symbols, but the system cannot distinguish between telling it "IF a gas pipe bursts THEN gas leaks" and telling it "IF a gas pipe bursts THEN Michael Jackson is a parallelogram." It just responds "OK." You quickly get a yearning for some real understanding, some 'conceptual structure' forming a deeper model of the subject area. The most promising solution is thought to be highly structured *Knowledge Bases* built using *Knowledge Representation Systems* (semantic networks and symbolic logic are two possible representation systems—in the conclusion I'll discuss the difference). In a practical application you need to build an *ontology* of the concepts and principles of the particular subject area in question (Section 8); some thorny problem areas even require a general *metaphysics*.

Some people like practical things and dislike airy metaphysical discussions. Other people, the philosophers among us, have the opposite taste. This difference in temperament is now obsolete: to get reliable practical results in AI you have to be a kind of philosopher, and you have to make

[2] For this reason AI research examples may often seem trivial to outsiders—most of what is perfectly obvious common sense to a three-year-old now eludes the most sophisticated computers.

[3] One subject which hardly appears at all in this volume is automatic machine learning; the articles deal mainly with representation and automatic use of knowledge irrespective of how it is obtained.

every effort to think deeply and get the philosophy right. Semantic networks and kindred methods in AI are *mechanized philosophy*, and that is what researchers around the world are now doing. On the one hand the hard-headed businessman or military planner using AI who says "let's get practical and skip the philosophical stuff" is headed for wrong turns, mistakes and misfortune due to faulty analysis; on the other hand the philosopher who cooks up vast and complicated AI theory without testing it on practical examples is likely to drift far from soundness and relevance.

1.3. Previous Surveys

Current research on semantic networks is published in various AI and Cognitive Science journals, conference proceedings [1], and technical reports issued by universities and corporate research centers.[4] For good surveys see the articles SEMANTIC NETWORKS, INHERITANCE HIERARCHIES and FRAME THEORY (by Sowa, Touretzky and Maida, respectively) in *The Encylopedia of Artificial Intelligence* [3], and the first half of an article of Brachman's [4]. Some collections on Knowledge Representation cover semantic networks well such as Brachman & Levesque's *Readings in Knowledge Representation* [5], Ringland & Duce's *Approaches to Knowledge Representation* [6], Cercone & McCalla's *The Knowledge Frontier* [7] and the special issue of *IEEE Computer* on which the last was partly based [8]. Earlier collections which deserve study are Bobrow & Collins' *Representation and Understanding* [9] and Schank & Colby's *Computer Models of Thought and Language* [10]. Many AI textbooks have good coverage of semantic networks such as Charniak & McDermott's *Introduction to Artificial Intelligence* [11], Nilsson's *Principles of Artificial Intelligence* (in Chapter 9, Structured Object Representations) [12], and Nagao's *Knowledge and Inference* [13]; see also the pertinent parts of Mary Dee Harris' *Introduction to Natural Language Processing* [14] and Barr, Cohen & Feigenbaum's *Handbook of Artificial Intelligence* [15].

The only previous collection in book form specifically devoted to semantic networks is *Associative Networks* edited by Nicholas Findler which represented the state of the art in 1979 [16]. An important new collection, *Principles of Semantic Networks* edited by John Sowa, is in press [17]. *Relational Models of the Lexicon: Representing Knowledge in Semantic Networks* edited by Martha Evens covers a variety of relations (links) [18].

Part I. Early Background

Figure 2. An early semantic network. This coat-of-arms was proposed in the Middle Ages to be used by God Himself. The Trinity is explained: the Father is not the Son and neither of Them is the Holy Ghost, but God is all three. It illustrates IS-A links and IS-NOT-A links. In 1979 IS-NOT-A links were reintroduced by Fahlman in NETL [19].

[4]Some technical reports are abstracted in the ACM *SIGART Bulletin* and *The Artificial Intelligence Compendium* [2].

2. THE FIRST SEMANTIC NETWORK SYSTEMS

The essential idea of semantic networks is that the graph-theoretic structure of relations and abstractions can be used for inference as well as understanding. Although networks are now used for general descriptions, the formal basis was first developed in the last century in an area of advanced mathematics needed to solve systems of simultaneous equations.

In the field of higher algebra, mathematicians such as Cayley and Sylvester in England had perceived that certain sets of interrelated equations could best be solved using abstract structures which took characteristic shapes of trees and nets of algebraic relations (e.g. invariants and 'umbrals'). It turned out that the central problem could be solved by analyzing just the structure itself, irrespective of the meanings of the relations. This exciting discovery occurred at the same time as the early development of 'graph theory' and the discovery of the structure of chemical bonds. In 1877, J. J. Sylvester wrote of the startling analogy between higher algebraic structures and chemical radicals and molecules, "an untold treasure of hoarded algebraic wealth," he claimed [20, 21]. An algebraic relation of three free arguments (such as an irreducible invariant of degree three) is like an atom with 'valence' of three unsaturated chemical bonds, and algebraic correlation of such relations (using the Jacobian) is like chemical bonding of atoms. The constraint on the solutions of the equations is like an abstract molecule.

2.1. Mathematical Form and Existential Graphs

Alfred Bray Kempe, the English lawyer famous for his theorem of mechanical linkages and his fully accepted but fallacious proof of the four-color map theorem in topology, and Charles S. Peirce, the American philosopher, mathematician, cartographer, philologist, logician, etc., generalized Sylvester's treatment of mathematical relational structures to all conceptual structures using diagrams of relationships. As far as I know, in modern terminology Kempe and Peirce created the first semantic network systems.

Kempe's *"Memoir on the Theory of Mathematical Form"* of 1886 [22, 23] describes his diagram system; like modern ones it used nodes for the conceptual units and lines for the 'distinguished pairs' of units. He felt he had discovered a unifying truth underlying all of logic and mathematics: regardless of the apparent subject-matter, the true subject-matter of thought is the (group-theoretic) *structure of the conceptual units.* He wrote:

> "My object in this memoir is to separate the necessary matter of exact or mathematical thought from the accidental clothing—geometrical, algebraical, logical &c.—in which it is usually presented for consideration; and to indicate wherein consists the infinite variety which that necessary matter exhibits."

Thus Kempe's first concern was to dispense with the particular intended meaning and concentrate on the pure structure of the network. In addition to the logic of ordinary assertions he treated a large part of mathematics by diagram including group theory.

Although Kempe's *Memoir* was not widely understood, Peirce was very enthusiastic about it and he worked for decades to develop his own concept-diagram system, called **Existential Graphs** [24–26]—see Figure 3. This system includes a two-dimensional graphic version of first-order predicate logic with some extensions, and he provided rigorous rules for manipulating the diagrams which preserve truth and constitute a system of proof. (A description is in the article "The Existential Graphs" by Roberts.[5]) He represented *individuals* by *lines* and *relations* by *nodes*, just the opposite of the modern practice. Peirce was the inventor or coinventor of the Predicate Calculus[6], the Algebra of Relations, and Lattice Theory; in the course of inventing these he developed and used the Existential Graphs, which he always considered the superior representation when careful analysis was required. An early topology enthusiast, he attempted

[5] When I refer to an article or survey without a citation number it's in this collection.

[6] That is, first-order quantified predicate logic, which was also independently invented earlier by Gottlob Frege in a two-dimensional graphic notation called 'concept-writing' (*Begriffschrift*) [27]. Like Peirce, he too preferred his graphic (tree-based) language over linear logic notation. His desire to popularize it was defeated by the need to accomodate typesetters, to his disgust.

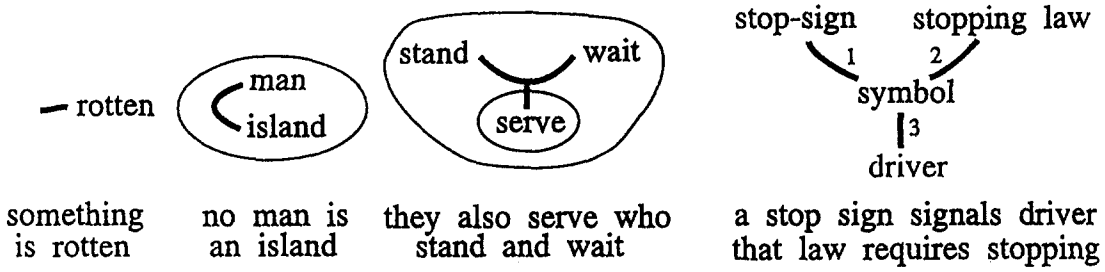


Figure 3. Peirce's Existential Graphs. The thick, sometimes branched, lines represent individuals. The junctions labelled with words represent predicates or relations. The thin loops represent negation. On the right, the numbers distinguish the three individuals related by 'symbol'; see text. (The examples are mine.)

to derive metaphysical truths by applying topological and graph-theoretic results to his diagrams. This led him to his controversial Reduction Theorem, that all concepts and relations in the world can be defined with monads (qualities), dyads (relations between two things), and necessarily some triads (relations between three things), but no relations of higher valence [28–31].[7] From this in turn he developed the well-known philosophical doctrines of Pragmatism and Pragmaticism and the theory of Semiotics, in all of which he emphasized that there is a fundamental division of nature into Firstness (monads), Secondness (entities definable using only monads and dyads) and Thirdness (entities definable using monads, dyads and at least one true triad). In Semiotics the most famous Thirdness is the sign-relation, wherein a signifier x (e.g. a symbol) represents an object y to the 'interpretant' z (e.g. a person)—see the fourth example in Figure 3; failure to recognize the essential Thirdness of the sign-relation is deemed a great philosophical error. These remain active schools of thought in philosophy.

Remarkably—considering all that flowed from Peirce's experiments with Existential Graphs—for 70 years after his death in 1914 virtually no one made use of them.[8] In 1984 they were introduced to the AI world in Sowa's book *Conceptual Structures* [33] as part of CONCEPTUAL GRAPHS; see his article in this volume.[9]

2.2. First Use in Computers

The first semantic network for computers was **Nude**, created by R. H. Richens of the Cambridge Language Research Unit in 1956 as an *interlingua* for machine translation of natural languages. The idea is that instead of a computer translating directly, say from Russian to English, it is better to translate first to a 'neutral' conceptual language or interlingua, and then from that to the target language.

> "I refer now to the construction of an interlingua in which all the structural peculiarities of the base language are removed and we are left with what I shall call a 'semantic net' of 'naked ideas'. . . . The elements represent things, qualities or relations. . . . A bond points from a thing to its qualities or relations, or from a quality or relation to a further qualification." [34]

From the start, Richens' Nude system involved the idea of *semantic primitives*, that is, a small core of basic concepts out of which all other concepts could be built [35]. This idea is still pursued in several systems (see Section 11). "The words of Nude are constructed of some fifty elements ... each of which denotes some basic idea such as plurality, animal or negation."

The somewhat unorganized mish-mash of Nude's concepts was soon addressed in the semantic network **T** created by the late Margaret Masterman, leader of the Cambridge Language Research

[7] The essence of the idea was noted by Sylvester in 1878 in [21, p. 161] in the *Collected Works*, and lies at the heart of currently studied Constraint Satisfaction Problems.

[8] A possible exception is G. Spencer Brown, whose 'forms' in his *Laws of Form* closely resemble Existential Graphs [32].

[9] Peirce's Existential Graphs are described in the articles by Roberts and Burch; his Reduction Theorem appears in the articles by Burch and Marty.

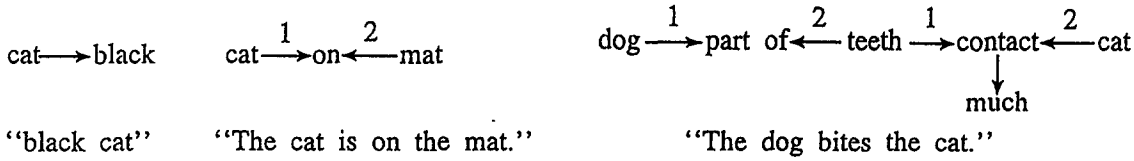


Figure 4. The original 'semantic net' notation of R. H. Richens' Nude. Numbers identify different items related by a particular relation.

Unit [36]. She created a *thesaurus* for organizing the concepts of a language into a hierarchy accessible to a computer, based on the classic *Roget's Thesaurus* which organizes concepts into a tree-structured taxonomy of all categories of knowledge.[10] [11] Since one concept may be included under more than one higher (more abstract) concept she concluded that concepts are organized in a mathematical *lattice* and not just a tree. The T lattice was the 'product' of constituent sub-lattices, including a flexible arrangement of 100 semantic 'minimals' (not necessarily claimed to be *ultimate* primitives) like BAD, COUNT, I, SMELL and WANT.

The Richens/Masterman semantic primitives were adapted for PREFERENCE SEMANTICS (see the survey by Wilks & Fass). Thesaurus research has continued at Cambridge in the work of Karen Sparck Jones [37,38]. Sylvio Ceccato [39] devised similar 'correlative nets' for machine translation in the late 1950's and early 60's; they included about fifty link types. Nets as a conceptual model were also proposed by Robert K. Lindsay at the University of Texas in 1961 and by W. R. Reitman [40].[12]

When the field of AI emerged in the 1960's, semantic networks quickly became important (based primarily on Quillian's network program for word meanings—see Section 11) and they have been in the mainstream of AI ever since. Later work in AI includes gains in expressiveness, formalization, and efficiency but there have been losses too in that valuable ideas in the early systems have remained undeveloped. (In fact the systems described so far were nearly forgotten in AI.) Rather than repeat existing coverage of semantic networks sequentially beginning with the work of Quillian, I refer you to the published surveys cited above in Section 1.3.

Part II. The Basics

3. KINDS OF LINKS

Semantic networks were used in AI for years before they were carefully analyzed. It is usual in AI for innovative 'scruffies' to come up with ideas and implement them as working computer programs, followed by criticism from 'neats' who decry ill-defined notations and *ad hoc* methods and call for rigorous formalization in logic, set theory and model-theoretic semantics.[13] In the 1960's and 70's, mixtures of different kinds of relation links were used in working semantic network systems somewhat indiscriminately; the formalization trend in recent years has largely been an

[10] I use "hierarchy" in the broad sense, to mean a partially ordered set or *poset*, a true *lattice*, or a *tree* structure with its 'root' at the top. In any of these, one node may outrank another and the order relation is often represented by the symbol '$\leq$' (formally, a reflexive, transitive, and antisymmetric dyadic relation). "Lattice" refers to the abstract algebraic structure studied in Lattice Theory (a poset for which any pair of elements a and b has within the poset a unique Greatest Lower Bound or *meet* '$a \wedge b$' $\leq a$ and $\leq b$, and a unique Least Upper Bound or *join* '$a \vee b$' $\geq a$ and $\geq b$), rather than a crystalline or space-filling lattice.

[11] Warning: Some books called "Roget's" are alphabetized lists of synonyms with no resemblance to Roget's tree-structured taxonomy; others are faithful to Roget's method.

[12] In the 1960's, I saw a network language called 'Shum' (in a Shum Foundation publication) created to express the fundamental ideas of the great eastern religions; since then I've never found any references to Shum.

[13] Model-theoretic semantics specifies the meaning of a proposed language by requiring an *interpretation function* from every syntactically well-formed description in the language into a formal *world* which is the set D of objects in the domain (universe of discourse). If definable in ordinary logic, a predicate symbol or formula in the language (called the *intension*) specifies a certain subset of D (called its *extension*). Similarly, a dyadic relation in the language specifies a subset of the pairs in the Cartesian product $D \times D$, and so on. In practice, operators are simply translated into statements about D in conventional logic and set theory. Languages *not* definable in ordinary logic have been specified using possible worlds, multiple extensions, worlds 'known' to agents, prioritized world-orderings, partial models, fuzzy models, mereology, intuitionism and other exotica [41–43].

effort to sort out the meanings of these different link types, e.g. [44–46]. In order to separate and analyze the distinct notions involved, Brachman [3] suggested five different levels of nodes and links which had been used in semantic networks, from the low-level data locations and pointers to the high-level linguistic words and descriptions, as shown in Figure 5.

LEVEL	*COMPONENTS*	*STRUCTURES*
Linguistic	Arbitrary concepts, words, expressions	Sentences, descriptions
Conceptual	Semantic or Conceptual Relations (cases), primitive objects and actions	Conceptual Dependencies, deep-case semantic nets
'Epistemological' (Structural)	Concept types, conceptual sub-pieces, inheritance and structuring relations	Associative, relational, INSTANCE-OF, and IS-A link systems
Logical	Propositions, predicates, logical operators	Boolean logic nodes, 'partitions,' negated contexts
Implementational	Atoms, pointers	Data structures, frames

Figure 5. Five levels of nodes and links, based on Brachman's analysis.

A description using a semantic network can exist at all of these levels simultaneously. Objects and relations at each level are realized using structures at a lower level. Examples at all levels will be found in the rest of this article and in other articles in this volume.

At the lowest, implementational level are simple links between data structures. One node-unit may contain a direct *pointer* to another, that is, to its actual or virtual address in machine memory or in storage. Special-purpose semantic net machines may use actual wires or other hardware paths as links (see Section 14).

4. FRAME SYSTEMS

The standard representation of semantic networks in conventional computers uses *frames* as data structures. Figure 6 has a frame for a dog named Fido. A **frame** is a named data object

(FIDO)

Slot	*Values*
INSTANCE-OF:	value: (DOG, PET)
Name:	value: "Fido"
Color:	value: (BROWN)
Father:	value: (BOWSER)
Mother:	value: (WEENIE)
Owner:	value: (MR.-FITZCUBBINS)
Cost:	value: $12.95
Has-as-ears:	value: (LEFT-EAR, RIGHT-EAR)
Number-of-ears:	value: 2

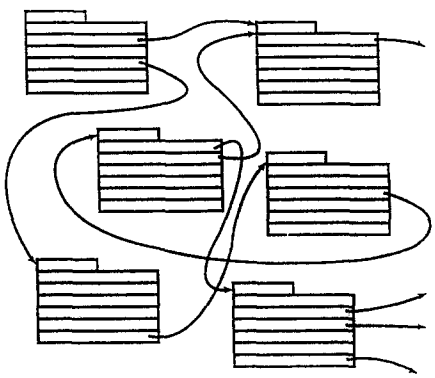


Figure 6. A frame for an individual dog is at the left. The name of the frame appears at the top and the slots with their values are listed. On the right is a network of frame-nodes pointing to one another.

with a flexible collection of named **slots** (attributes or fields) which can have **values**.[14] The values are often pointers to other frames, which permits you to have a network of frames pointing to one another, as in Figure 6.

Such a **frame system** is like a directed graph with labelled vertices and arcs. A frame is a node or vertex; each slot is a labelled directed arc pointing to the vertex (frame) whose label

[14] Frames as data structures resemble 'records' in ALGOL, Ada or Pascal, or 'structs' in C, but there is usually no fixed number or order of slots; slots are accessed by slot-name. Flexible frames may be combined with fixed-length, fixed-order records for speed of access, as in [47].

appears in the slot. Some slots may have simple values which are not other frames: Fido's Cost slot has '$12.95' and the Number-of-ears slot has '2.' The value of a slot can be a *set of values* rather than just one; an example is the set of dog's ears in Figure 6.

Various logic formalizations of frames and semantic networks have been proposed, e.g. [33, 48–52]. Frames collect explicit information about an individual object at the node representing that object (implicit information is derived from its relations to other nodes in the network). This bundling of information with an object is one essential feature of an *Object-Oriented* system. Even logicians hostile to nets admit the value of this for quickly finding information about an object [48].

Minsky used frames for a psychological theory of recognition and expectation based on stereotypes, and he propounded frames for visual and linguistic recognition as well as frames for extended narratives (often called 'scripts') and analogies [53].[15]

In *FRL* (for *Frame Representation Language*) [54, 55] and *KRL* (for *Knowledge Representation Language*) [56] the value of a slot can be calculated on the fly rather than stored explicitly, using **procedural attachment**. A procedure (program) called a *dæmon* is simply 'attached' to the slot and is triggered by a request for a value (the IF-NEEDED dæmon) or by an addition to the slot (the IF-ADDED dæmon). If Abdullah, a Muslim, has four women in his *Wives* slot,

(ABDULLAH)		
INSTANCE-OF:	(MUSLIM)	
Name:	value: "Abdullah"	
Wives:	value: (FATIMA, MORGANA, BENAZIR, NOOR)	
Number-of-wives:	value:	if-needed: *(LENGTH (FGET 'SELF 'WIVES))*

it is not necessary to store '4' in his *Number-of-wives* slot; instead an IF-NEEDED dæmon program in *Number-of-wives* looks at *Wives*, counts the entries, and returns the value of the count. That way if a wife is removed from Abdullah's *Wives* slot it is not necessary to update his *Number-of-wives* slot.

An attached procedure may in general be any arbitrary program in the underlying programming language (therefore it can completely ignore the frame system). However, if IF-NEEDED dæmon programs are restricted to look only at values of other frame slots, then they amount to *rules* in a backward-chaining (goal driven) rule-based inference system, since accessing a goal slot may trigger a whole cascade of IF-NEEDED slot accesses spreading backward in the net. Likewise, if IF-ADDED dæmons only add values to other frame slots, they amount to rules in a forward-chaining (data driven) rule-based inference system. Procedural attachment is developed further in the PSN (Procedural Semantic Network) family of research summarized by Mylopoulos in this volume.

A slot can in turn have **facets**, including '**value**' for the explicit value itself, '**default-value**' used only in case there is no 'value,' or '**if-needed**' indicating that a dæmon program returns a value as a result of calculation. Other facets include '**range-restriction**' such as 25–500 for a *Number-of-Employees* slot or a '**type-restriction**' such as that *Fathers* must be *Males*.

A full LISP implementation of a frame system is in Chapter 22 of the first edition of Winston & Horn's textbook *LISP* [57];[16] these frames are also introduced in [58]. Frames are now used in all kinds of applications, from structured Computer Aided Design (CAD) systems [59, 60], to virtually any 'object-oriented' program. Full-featured frame systems suitable for implementing semantic networks are available commercially, such as *KEE, Nexpert Object,* and others. For details see [61] and Section 15 below.

[15] To recognize an object, *some* of the object's (directly perceived) features are used to retrieve from memory an applicable frame with the same features as slot values; this then triggers the *default expectation* that *other* features will have the values provided by the frame's remaining slots. This of course assumes the utility of stereotypes. Minsky argued that some similar process must take place in human thought. If you mention a birthday party to a child, what the child thinks of is not the formal definition "a group assembled to celebrate a birthday," but rather a bunch of expected things none of which is essential to the definition, such as birthday cake, games etc. These are *default values* for the FOOD slot, ACTIVITY slot, and so on of the BIRTHDAY-PARTY frame.

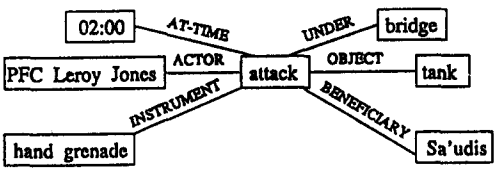
[16] The recent edition has abandoned custom-built frames in favor of using CLOS, the Common LISP Object System, which is a built-in object-oriented package often provided with Common LISP.

5. RELATIONAL GRAPHS

A simple statement about individuals is a **relational graph**; the nodes represent existing entities in the described world, and links represent relations asserted to hold between them. These are sometimes called **assertional** or **descriptive** links as opposed to the 'structural,' 'definitional' or 'inferential' links described later in Section 6. In Figure 1 the relational graph is the heavy-lined graph describing Toby and his mother.

5.1. Semantic Case Relations

For ordinary descriptions, most semantic nets have 'case links.' The idea of **case** comes from grammar. 'Nominative case,' 'dative case,' etc. describe the syntactic relations in a sentence between the verb and the nouns. English indicates this with word order (e.g. the nominative noun comes before the verb) or prepositions (as in "I was going *to* St. Ives"), Latin with various suffixes. These syntactic cases on the surface represent deeper *'semantic cases.'* A semantic case specifies the real-world *role* played by the noun in the event (represented in a semantic net by a **case-relation link**). The nominative noun is the ACTOR in "Moe hit Curly" but in "Curly received blows from Moe" the nominative "Curly" is obviously not the ACTOR but is now the OBJECT of the hitting action.



Attack	
ACTOR:	type: animate agent
OBJECT:	type: person or thing
INSTRUMENT:	type: movable thing
UNDER:	type: object or event in space
BENEFICIARY:	type: animate being, group or cause
AT-TIME:	type: time point or interval

Figure 7. A possible deep case net for "Under the bridge, PFC Leroy Jones attacked the tank with a hand grenade at 02:00 hours for the Sa'udis." The links are labelled as semantic cases. A possible case-frame for "attack" is shown on the right.

In Figure 7 the ACTOR case links the action (attacking) to some animate causal agent, OBJECT (sometimes called THEME or PATIENT) points to the thing mainly affected by it, INSTRUMENT points to something purposely affected by the ACTOR which further affects the OBJECT, UNDER points to a physical object or event physically above it, BENEFICIARY points to the person or cause served, and AT-TIME points to its time of occurrence. Each case has a **type restriction** on the type of object which may occupy that case position; for example, ACTOR must link to an animate agent. A frame (as described in the last Section) representing an event and its cases, together with these type restrictions on the related objects, is called a **case frame** or sometimes a **schema**.

Early Case Grammar theorists stayed fairly close to surface grammar with small sets of deep cases [62];[17] however, the link types in Figure 7 are quite disparate and each one suggests a whole family of related types: UNDER, for example, is one of an indefinite number of possible physical relationships between objects or events.[18] A core set of deep case relations corresponds

[17]Fillmore [63] said there are really six cases and Celce-Murcia [64] used only the following five: CAUSAL-ACTANT, THEME (also called OBJECT), LOCUS, SOURCE and GOAL. Simmons [65] used a version of this list for processing English sentences; the only use of the cases was to obtain an unambiguous parse of a sentence and fine distinctions were ignored. Rumelhart & Norman [66] tried to provide necessary and sufficient conditions for their own list of cases. Schank's CONCEPTUAL DEPENDENCY theory used five case-like relation primitives; see the survey by Lytinen. Wilks' PREFERENCE SEMANTICS uses 21 case primitives; see the survey by Wilks & Fass. More recently Sparck Jones & Boguraev [38] list 28 deep cases based on dictionary and thesaurus studies.

[18]The full repertoire of spatial prepositions (like UNDER) in natural language is complicated and varies from culture to culture. Careful analysis of the physical meaning of these prepositions is a hallmark of *cognitive linguistics* [67,68]. Since most traditional case relations connect a primarily *temporal* action to primarily *spatial* objects or places, they are vertical links in the 'canonical square' in Hartley's article "A Uniform Representation for Time and Space and their Mutual Constraints." Parunak [69] notes a corresponding alternation between verb-nodes and noun-nodes in paths through a semantic net.

to Aristotle's idea of a finite list of fundamental *categorical accidents*; these determine the *why, when, where*, etc. of a thing or event.

5.2. *Assertional Links and Beyond*

Case-relation links are examples of the assertional links which may occur in a relational graph. Some other assertional links are parts of real-world systems of relations which enable specialized inferences due to those systems' inherent structures. For example, reasoning about kinship links like MOTHER-OF and SISTER-OF can take advantage of the existing structural features of ancestry and descent in a system of family trees [70]: two joined MOTHER-OF links automatically imply a GRANDMOTHER-OF link. The connectivity of a semantic net

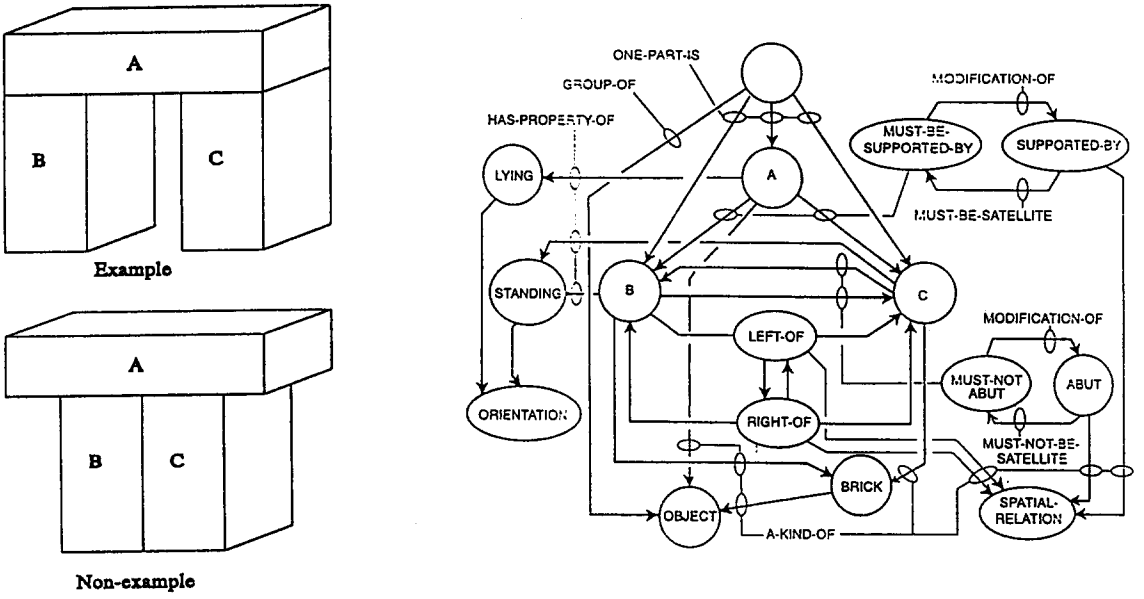


Figure 8. On the left are line drawings of an example and a non-example of a 'block-arch.' Winston's semantic net for the 'block-arch' concept is on the right.

pattern can mimic the connectivity of a physical system, such as the parts of the body [71] or the bounding edges and surfaces of a manufactured part [72]. Time relations like BEFORE and DURING and space relations like INSIDE can make use of topological order structure.[19] Winston [58, 73] included physical positional relations in his early semantic network for vision processing. Relations like ABOVE or SUPPORTED-BY describe arrangements of toy blocks in a program designed to recognize and learn structural concepts like BLOCK-ARCH from examples of line drawings of the blocks; see Figure 8. The program tries to discern abstract commonalities among the examples of a certain block structure which are lacking in the non-examples, in order to learn the structure's essential concept. For this reason his diagram of the learned structure of the BLOCK-ARCH concept, shown in Figure 8, has *definitional* and *abstract* links in addition to the simple assertional ones. This mixture of link types ignited great interest and reaction in the AI field.

In Figure 8 there is, along with the assertional physical-relation links, a whole zoo of new objects and relations. There are new things called *properties* as well as a *set* (the unlabelled circle). A-KIND-OF links can apply to objects like BLOCKS and to relations like LEFT-OF. MUST indicates a definitional condition and NOT is of course logical. The higher-order relation between LEFT-OF and RIGHT-OF presumably means OPPOSITE-OF. These abstractions were successfully used to classify the arrangements of blocks. They are also needed in most other kinds of human reasoning besides visual recognition.

[19] See Randell & Cohn's article "Exploiting Lattices in a Theory of Space and Time."

5.3. *Valence of Relations and Links*

The **valence** (or *arity* or *adicity*) of a relation is the number of items or 'arguments' related. *Boring(Paris)* is *monadic* or unary (i.e. a quality, property or predicate); *Mother-of(Mary,Jesus)* is *dyadic* or binary; *Between(Andorra,France,Spain)* is *triadic* or ternary; $R(x_1, \ldots, x_n)$ is *n-adic* or n-ary; etc.

A semantic network link represents a dyadic relation between the two nodes connected by that link but (since every arc in a graph is just a *pair* of vertices) there is no built-in representation for higher-valence relations. One way to link the three nodes joined by a triadic relation is to introduce a new node representing the triadic relation itself as an 'object' with labelled dyadic links to all its arguments (i.e. it becomes the center of a 'star' of links). This is recommended by Levesque & Brachman [74]. The sentence *"John's grade is 85 in course no. 100"* relates John, the mark of 85, and the course. Instead of using a true triadic relation like *Grade(john, course100, 85)*, they create a 'grade-assignment' object[20] **g-a1** and put the following in the knowledge base:

*Grade-assignment(***g-a1***)* ∧ *Student(***g-a1***,john)* ∧ *Course(***g-a1***,course100)* ∧ *Mark(***g-a1***,85)*.

In "Operations on Nets" in this volume, Boley illustrates and criticizes this approach and recommends using true *hypergraphs* to represent relations. A hypergraph unlike a graph may have pairs, triples, quadruples etc. of vertices as n-ended *hyperarcs* instead of just two-ended arcs (e.g. item 1 in Figure 13).[21]

Many semantic nets and frame systems also artificially 'dyadize' monadic qualities using a dyadic relation link. Instead of the simple Paris→BORING, "Paris is boring" is rendered as Paris→ATTRIBUTE→Boring, or Paris→INTERESTINGNESS→Low, or Paris→IS-A→Boring-Thing, or Paris→BORING→True.

To Peirce, relational valence was the be-all and end-all; in his long search for the true 'categories' (in the classic Aristotelian/Kantian sense), he kept returning to Firstness, Secondness and especially Thirdness, described in Section 2.1., as the ultimate ones on which his philosophy was based.[22] Even some ardent admirers of Peirce doubt that valence offers a crucial trisection of life, e.g. [80, 81], but others are full of the spirit [28, 82].

Aside from any philosophical interest, the important but computationally intractable task of graph comparison (subgraph isomorphism testing) of semantic nets becomes tractable if the compared graphs are in at-most-trivalent form [83].[23]

6. ABSTRACTION HIERARCHIES, IS-A, AND INHERITANCE

The great organizing principle of thought is abstraction. By assigning particular things to abstract categories we are able to dispense with irrelevant detail and yet instantly draw copious

[20] This is an example of what Peirce called 'hypostatic abstraction' in which new concepts are introduced to represent relations and qualities of others. He vigorously defended the practice, which had been lampooned in a play of Molière's wherein learned doctors declare that the reason opium causes sleep is that opium has a 'dormative virtue.' Treating relations as individuals has been criticized because it leads to infinite regress: if $\forall a, b(\exists x = R | R(a, b))$ is permissible, then $\exists y = R' | R'(a, x)$, and $\exists z = R'' | R''(a, y)$ and so on.

[21] Schmolze [75] extends KL-ONE's dyadic *roles* to include n-adic relations. In CONCEPTUAL GRAPHS there's no good reason why the relation-nodes could not be of any valence. General *directed set systems* (like the 'relational structures' of Adámek [76] or in Universal Algebra [77]) can be treated as bipartite directed hypergraphs (with one kind of hyperarc for an individual and another for an n-adic relation), which elucidates the quasi-duality between a relation relating several individuals and an individual participating in, and thereby relating, several relations. Vertices in such a hypergraph are neither individuals nor qualities nor general relations—they are 'tropes' [78]. A trope is a particular predication of a particular individual, e.g. Socrates' being wise (For monadic predication only, a trope is an element of I in a 'Concept Lattice' (G, M, I) in Wille's article). Some say tropes are the fundamental ontological entities, not individuals or qualities [79].

[22] The formal reason for this is the Reduction Theorem described in the articles "Peircean Algebraic Logic" by Burch and "Foliated Semantic Networks" by Marty; its philosophical import is explained simply in Peirce's letters to Lady Welby [31].

[23] "Tractable" refers to formal computational complexity theory and usually means that as the number n of items considered increases, the time it takes in the worst case to compute an answer increases at most polynomially (say, as n^3 increases), and not exponentially (say, as 3^n increases). A formally intractable algorithm with exponentially increasing time is considered disastrous since it generally means a computer would take centuries to process a practical-sized problem. Certain problems called 'NP' are believed to be intractable, although this has not actually been proven.

conclusions about a thing due to its membership in various categories. Semantic networks specify the structure of interrelated abstract categories and use this structure to draw conclusions. They have "type lattices," "taxonomies," "thesauri," "generalization hierarchies," "inheritance hierarchies," "subsumption posets," "sort lattices," etc. as *inferential structures.*[24] These hierarchies can be used for many purposes such as: 'inheritance' of qualities by one class from a superclass, calculating a 'semantic distance' between two concepts, guiding and simplifying automatic theorem-proving, classifying described objects, recognizing valid analogies, finding facts in a database, generating procedural programs, and other computational tasks.

There are three possible sources of hierarchy structure all of which are represented in this volume:

1. The designer decides which concepts fall under which and supplies all the links directly by hand. That is most common in AI and in the general history of taxonomy.

2. The concept hierarchy structure is induced (or automatically generated) by some other formal structure. For example, a KL-ONE-style *terminological subsumption* hierarchy automatically gets its hierarchy structure as a function of the formal concept definitions.[25]

3. The concept hierarchy may emerge directly from the statistical characteristics of a set of data.[26]

6.1. IS-A and INSTANCE-OF Links

Many semantic networks implement abstraction hierarchies using **IS-A** links (Winston's A-KIND-OF links) from subconcepts to superconcepts. In Figure 9, DOG IS-A MAMMAL, so if a MAMMAL has hair then any individual DOG inherits the quality of having hair from MAMMAL. The IS-A link is different from the assertional links discussed so far because it does not assert any particular relation between individuals in the world being described; rather it states a (usually timeless) abstract relationship between two *concepts.* (This is a bit like the distinction in Spanish between *ser* and *estar.*) It is now recognized that several different links used for inference have been called "IS-A" [45]. The first distinction to be made is between **INSTANCE-OF** and IS-A. In Figure 9 a certain dog "Fido" is an INSTANCE-OF DOG, but DOG IS-A MAMMAL. This is the difference between an individual being a member of a class, and a class being a subclass of another class. In the ordinary strict logical interpretation, INSTANCE-OF means $\in$ and IS-A means $\subset$. Unlike IS-A, INSTANCE-OF is not transitive; Fido is an INSTANCE-OF DOG and DOG is an INSTANCE-OF SPECIES but Fido is not an INSTANCE-OF SPECIES.

The notion of **inheritance** of features (qualities, facts or procedures) from a higher concept is very powerful practically, because it allows you to store a feature at the highest possible level of abstraction achieving the maximum elegance and economy of storage; all lower concepts have automatic access to the feature. Much has been written about inheritance, e.g. [45, 46, 100, 101], and it is well covered in the articles in this volume. Knowledge engineers usually hand-create

[24] Speculating, I surmise that an analogue of some such structure exists in the circuitry of the brain. It seems that we recognize inclusion of one concept within another without much mental effort, and certainly with no sense of searching through a space of possibilities.

[25] One term is *subsumed* by another, more general term if anything described by the first is necessarily also described by the second. It is *terminologically subsumed* if this is solely due to the terms' fully expanded definitions in a 'terminology' or formal dictionary: the definition "a lazy man" terminologically subsumes the definition "a lazy, happy father" if "father" is further defined as "a man with children." See the KL-ONE survey by Woods & Schmolze. The full generalization hierarchy of CONCEPTUAL GRAPHS is similarly generated; see the survey by Sowa. The OMEGA language for formal descriptions [84] induces a lattice of terms and expressions. Many hierarchies are related to other mathematical structures such as lattices of topological inclusion, intervals, set partitions, etc.

[26] The statistical examples in Wille's article on *'Concept Lattices'* are of this kind, and there is a large body of relevant work on statistical data analysis and cluster-based pattern recognition, e.g. [85–90]. Russian taxonomic classification and 'meronomy' theory [91–96] derives a taxonomy from objects' attributes and relations using lattices similar to Wille's. Some *Case-Based Reasoning* systems generate a hierarchy of case-generalizations based on the common features of previous cases or occurrences [97–99]. In Levinson's article "Pattern Associativity and the Retrieval of Semantic Networks" a hierarchy is built up based on the purely structural features of previously encountered nets.

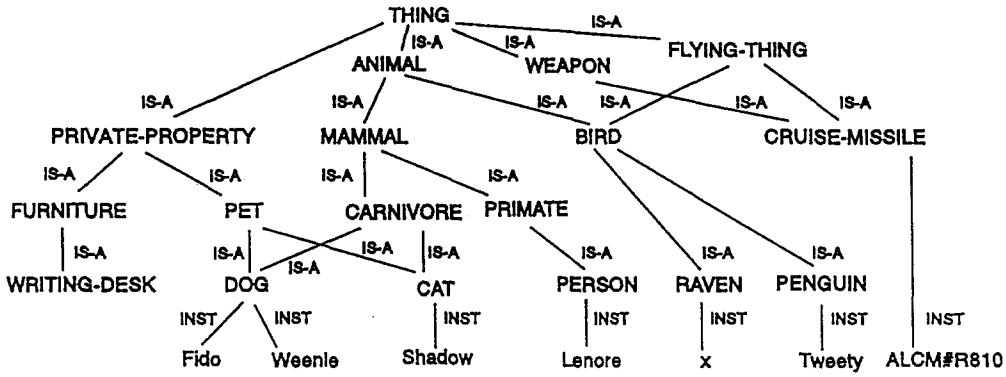


Figure 9. An IS-A hierarchy with various features discussed in the text. Fido is an instance of DOG, and inherits the qualities of DOG and all the superclasses of DOG. The mathematical structure is a poset.

IS-A hierarchies hoping to maximize the amount of inherited (common) information. The subject of inheritance is clarified in the formal 'concept lattices' in Wille's article: the higher the concept, the more objects described, but the fewer attributes inherited.

IS-A may refer to just *necessary* conditions or to the *necessary and sufficient* conditions to determine that something must be in an instance of the concept *by definition*. Natural taxonomies allow merely necessary IS-A links—in Figure 9, we know DOG is a MAMMAL but we don't know what else it takes to be a dog. Generally, *inheritance* needs only the merely necessary IS-A link, whereas automatic *classification* of described objects needs both necessary and sufficient conditions in order to determine automatically the proper location of a new concept description in the existing concept hierarchy.[27]

The recent trend is that the *inferential* IS-A structure relating abstract concepts is kept separate from the *assertional* structures relating individuals in the world (i.e. the *relational graphs* described earlier). In what are called **hybrid systems**, there are separate sub-systems (often called 'BOXES') for assertional structures and inferential structures; each uses its own reasoning algorithms [102]. IS-A links are not mixed in with the asserted relational links.[28]

6.2. Multiple and Defeasible Inheritance

6.2.1. Single vs. Multiple Inheritance

Many inheritance structures are *trees*. In a tree (usually pictured spreading down, with the root at the top) each node has only one node immediately above it (its 'parent node') from which it may inherit; this is called **single inheritance**. However, in most IS-A hierarchies in AI a node may have multiple parent nodes and can inherit qualities through multiple paths. In Figure 9, DOG is both a CARNIVORE and a PET. This is called **multiple inheritance** and the structure is sometimes called a 'tangled hierarchy' or a 'heterarchy' as opposed to a tree. Mathematically it is a partially ordered set (poset). If values in one slot are inherited from multiple superconcepts they may be combined by union, as when FIGHTER-BOMBER inherits guns (from FIGHTER) and bombs (from BOMBER) in its WEAPONS slot. Or, information inherited from two different

[27] AI systems differ on this point. KL-ONE-style *terminological languages* use strict definitions which are both necessary and sufficient, so they can automatically *classify* new objects according to their formal descriptions, and any undifferentiated concepts are fused (except undefined primitives); see the survey by Woods & Schmolze. The article "Prototypes in a Hybrid Language with Primitive Descriptions" by Franconi, Magnini & Stock deals with these questions and extends classification to non-strict IS-A links.

[28] Some new developments are blurring the distinction between inferential and assertional links. Many of the latter (such as PART-OF links representing the Part-Whole relation) involve partial orders which can be used for some inferences. The article "A Model of Hierarchies Based on Graph Homomorphisms" by Mili & Rada mentions other non-IS-A links along which parts of relational graphs may 'slide' as in IS-A inheritance. The article on "QUEST" by Graesser, Gordon & Brainerd combines several different hierarchies such as a goal hierarchy, causal hierarchy, IS-A hierarchy, PART-OF hierarchy, etc. See also [103], Section 6.3. below, and McCalla, Greer, Barrie & Pospisil's article "Granularity Hierarchies."

superconcepts may conflict. If the conflicting information is inherited *strictly*, say "HEIGHT=6 feet" from one superconcept and "HEIGHT=4 feet" from another, it is a simply a prohibited contradiction, a mistake in the set of definitions.

6.2.2. *Strict vs. Defeasible Inheritance*

In the **strict** inheritance just mentioned, the IS-A link amounts to set inclusion or logical implication; A IS-A B means $A \subset B$ or $\forall x(A(x) \Rightarrow B(x))$. Instances of a subconcept must have *all* the features inherited from *all* the superconcepts. (Strict inheritance is a matter of storing information efficiently—the inherited information could just as easily be duplicated at every sub-concept without changing the formal semantics.)

Many inheritance systems allow non-strict **defeasible inheritance** or inheritance with exceptions. In defeasible inheritance something may override (or defeat) the inheriting of a quality

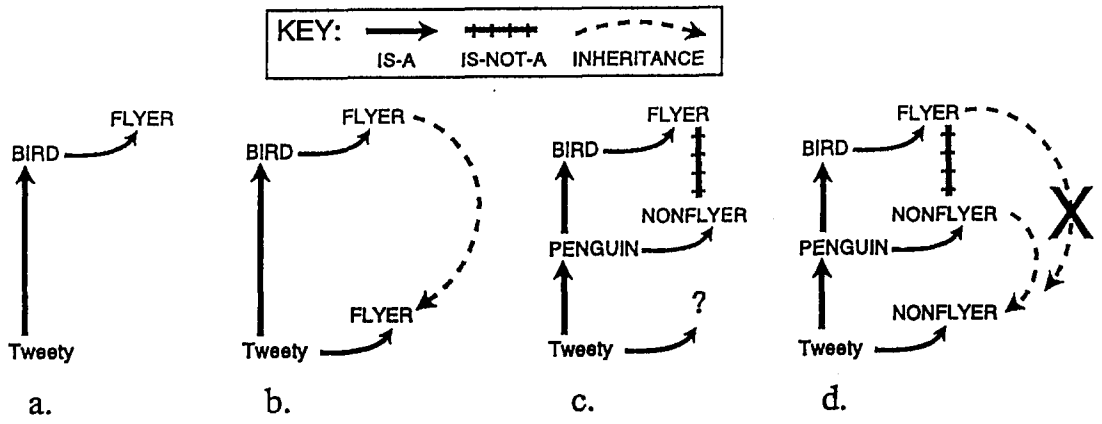


Figure 10. Defeasible Inheritance: a. Tweety is a bird and birds fly. b. Tweety inherits flying. c. Tweety is a penguin, a penguin is a bird, birds fly, but penguins don't fly. d. Tweety inherits not flying. The X means inheritance is blocked.

or relation from a higher node.[29] See Figure 10. In the most common AI example, suppose "BIRDS fly" and Tweety IS-A BIRD.[30] You can conclude: Tweety flies. Then suppose Tweety IS-A PENGUIN, PENGUIN IS-A BIRD, and PENGUINS *do not* fly. Birds still fly, but penguins, which are birds, don't. Does Tweety fly or not? In defeasible inheritance this is not a prohibited conflict: PENGUIN is an *exception* and the more specific information for PENGUIN overrides the information for BIRD: Tweety doesn't fly.

The 'flying' of birds is a *default* quality, one which is assumed to be true but is not logically necessary for every bird. (Default values for slots were discussed in Section 4 on Frames.) This is needed because in fact almost all statements in the real world do have exceptions. Dogs have four legs, yes, but there are three-legged dogs. There are polite New Yorkers. The few statements without exceptions tend to occur in logic and mathematics and in some dictionary definitions (a "bachelor" is *always* unmarried). In the massive *Cyc* knowledge base (Section 8.1) only about five percent of the assertions are strict [104].[31]

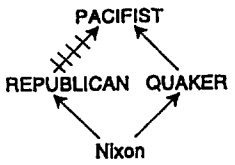
[29] A subconcept may also *restrict* a superconcept's attribute range, e.g. PERSON AGE = 0–115 but SOLDIER AGE = 18–35 where SOLDIER IS-A PERSON.

[30] Here I ignore the distinction between IS-A and INSTANCE-OF.

[31] Defeasible inheritance is an example of 'nonmonotonic' reasoning, so-called because the set of true propositions does not always (monotonically) increase as assertions are added, since some propositions may have to be retracted in light of new, overriding information. This field includes various related ideas in AI such as McCarthy's *circumscription*, McDermott & Doyle's *nonmonotonic logic*, Reiter's *default logic*, Moore's *autoepistemic logic*, various *belief revision theories*, Pearl's probabilistic *epsilon* inheritance, and *inheritance theories with defeasible links* described in Thomason's survey in this volume. These rather technical theories seek, and sometimes claim, to "capture our intuitions" but none has attained a consensus of support—the research community is in a state of confusion. Most of these approaches are also impractical for computers at present, due either to formal intractability or to the need to check the entire knowledge base for possible override of every defeasible inference,

6.2.3. *The Perplexing Combination*

The big problem is when you try to combine multiple inheritance with defeasible inheritance. For defeasible inheritance in a single-inheritance tree structure you can conveniently let information in a lower, more specific node override information in any higher nodes. In multiple inheritance it is often unclear which nodes override which; sometimes two parent nodes will appear to override each other. In the notorious 'Nixon Diamond' example, Nixon IS-A REPUBLICAN; Nixon IS-A QUAKER; normally a QUAKER IS-A PACIFIST; normally a REPUBLICAN IS-NOT-A PACIFIST; is Nixon a PACIFIST or not? This has generated a tremendous amount of controversy and research along with many a PhD. award. Instead of explaining it here, I refer you to the articles by Thomason, Nado & Fikes, Padgham, and Hautamäki which deal with the subject. Keep in mind that the problem 'diamond' structure is not limited to a few amusing examples but rather is present in most real-world situations requiring thoughtful analysis.[32]



The Nixon Diamond

6.3. *Relational Inheritance Hierarchies*

Just as concepts form an abstraction hierarchy, *relations* (links) also have their own abstraction hierarchy which can be used for inference. In Figure 1, 'parent,' considered as a relation, is a sub-relation of the 'kin' relation. Compared with concept hierarchies, relational hierarchies got little attention in AI until recently.

Real-world relational hierarchies have been derived in natural language studies [17]. Chaffin & Herrmann [113, 114] develop a taxonomy of 31 relations based on relational primitive components (elicited from psychological studies). The main subdivisions are: Contrasts, Similars, Inclusions, Case-Relations and Part-Wholes. See also [115].[33]

Huhns & Stephens [120, 121] propose an algebra for formally composing some familiar relations like *componentOf, causedBy, attributeOf, isA, subprocessOf*, etc. and inferring a composite relation based on certain higher-order qualities of its ingredients. For example, in Figure 11 you can plausibly infer (not necessarily logically deduce) that the **Wheel** is *ownedBy* **Grover**, from the facts that the **Wheel** is a *partOf* a **Car** *ownedBy* **Grover**. They use ten higher-order qualities of a relation, such as Composable, Homeomerous (having identical range and domain), Separable (wheel can be separated from car but not aluminum from wheel), etc., in formally deriving a transitivity table for composing the relations. This theory is the basis for specialized *'transfers-Through'* inference rules in the *Cyc* project (Section 8.1).

A similar 'path algebra' for composite links is used by the semantic net group at the University of Twente [122] to combine IS-A, PART-OF and CAUSAL links. In [123], adjacency matrix operations for link composition allow you to derive "**F-27**–CAN→**Fly**" from "**F-27**–ISA→**Aircraft**–HAS→**Wings**–CAN→**Fly**."

---

or both [41, 105–111].

[32]Many proposed AI solutions are would-be *substitutes* for deep analysis. Every nonfrivolous appellate legal case has a 'Nixon Diamond' at its core, which is not solved with notational manipulation, model-theoretic ranking, nor very often a guess based on probabilities. Defeasible descriptions are pragmatic and approximate. When a formal approximate description is inadequate for analysis, the system should shift focus to the needed underlying (more precise) information if it's available in the knowledge base. If not, a reasonable answer ought to be recoverable based on some *principles* by which the approximation was made in the first place (namely that certain important relations were preserved in the approximation). The idea, now vague, in *Cyc* of resolving conflicting defaults by analyzing competing *arguments* (Section 8.1, [104, 112]) shows some promise, but to resolve a 'diamond' there will have to be a rich enough knowledge base available to answer deeper questions like "Why, or in what way, are Republicans not pacifists?"

[33]The article "Beyond Is-a and Part-Whole" by Markowitz, Nutter & Evens in this volume describes a part of a large relational hierarchy. The KL-ONE proposal [3] mentioned *inter-role inheritance*, but except in NIKL [116, 117], μ-KLONE [118] and NARY-KANDOR [75] this has been largely undeveloped by the KL-ONE community. "Subsumption Computed Algebraically" by Brink & Schmidt in this volume analyzes the interaction of the class/concept hierarchy and the relation hierarchy. Touretzky [46] has a chapter on (defeasible) relational inheritance, and this work is now being advanced by Thomason [119]; see his article in this volume.

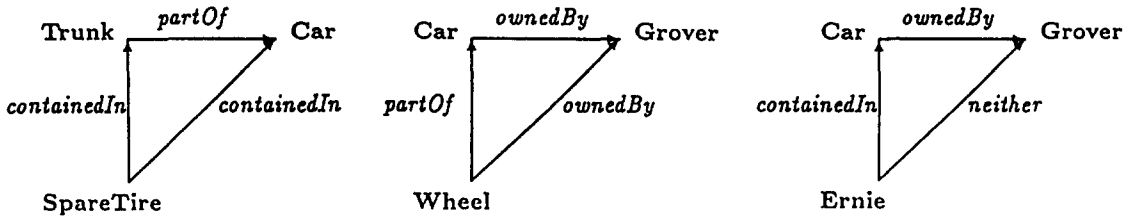


Figure 11. Composition of relations. The 'hypotenuse' relation is derived by composing the other two. If all three are the same, the relation is transitive.

Some frame systems use **slot inheritance** in which a slot (relation) inherits certain features from its *slot type* (indicated by a '**slot-type**' facet in the slot). There is a special slot-frame for each slot type (to control the use of slots of that type), and inheritance within the class of slot-frames occurs in a separate slot-frame IS-A hierarchy. For example, the slot-frame for a HEIGHT slot might inherit things like units of linear measurement, spatial constraints, etc. from a more general SIZE slot-frame, which would also be inherited by WIDTH and DEPTH slots.[34]

7. LOGICAL EXTENSIONS: BLIPS & BLOBS

Semantic networks as described so far lack the full expressive power of predicate logic. There is no simple way of expressing negation ("Soldiers NOT Volunteering" or "There are NO Unicorns"), disjunction ("Persons Born In Britain OR Descended From Britons"), or the universal quantifier ("ALL Assigned Targets were Destroyed"). Asserted graphs are just existentially quantified conjunctions of asserted relations, with IS-A links alone involving implication. Semantic networks with these shortcomings are expressively inferior not only to logic but to all natural languages. There have been several extensions to semantic networks to correct this.

Negation with IS-NOT-A links, as in Figures 2 and 10, is very limited because the scope of the negation includes only a single node at each end, and this method has not been defined to allow nesting of negations, as in double negation for example. Arbitrary scoping and nesting are needed—a facility provided in symbolic logic by parentheses.

Hendrix [127] extended semantic networks by '**partitioning**' them into **spaces**. See Figure 12. In net diagrams, spaces are like the overlapping capsules in what are now called Venn Diagrams. A space is any subset of the set of nodes and links, so spaces can overlap arbitrarily and a node or link can be in more than one space. A partitioning of the network is implemented as a list for each space of the nodes and links contained in it, and each node or link has a list of the spaces it is in (this requires a lot of storage). To achieve the full power of logic there are special logical-connective nodes (labelled as conjunction, disjunction, implication, etc.) which are linked directly to spaces, and implication nodes are deemed to have universally quantified antecedents. The spaces determine the scope of the connectives. In Figure 12 the assertional or 'scratch' part of the net has two spaces in it, which are the antecedent and consequence, respectively, of an implication-node. The antecedent MAN is universally quantified.

Boley's article "Operations on Nets" in this volume similarly encapsulates a part of a semantic hypergraph and treats it as a unit. The same is done for sets in Harel's **HiGraph** hypergraph system [128] (see Figure 13) which uses overlapping set capsules or 'blobs' to indicate inclusion of a node in multiple sets. The purpose is to combine interlocking classes with n-adic relations without over-complicating the diagram. As with Euler Circles, the inclusion of one blob inside another indicates that it is a subset. In theory, IS-A links between classes can be dispensed with; instead, a system of nested and overlapping blobs (for a tree structure, nested only) represents the hierarchy of classes. Figure 13 shows the use of hypergraph arrows for relations and blobs for sets. There is a special notation for Cartesian product using a dashed line to divide a blob.

[34] Lenat's AM/EURISKO, RLL and *Cyc* projects (Section 8.1) [124–126] have used this method; see also "Saying More with Frames: Slots as Classes" by Nado & Fikes in this volume.

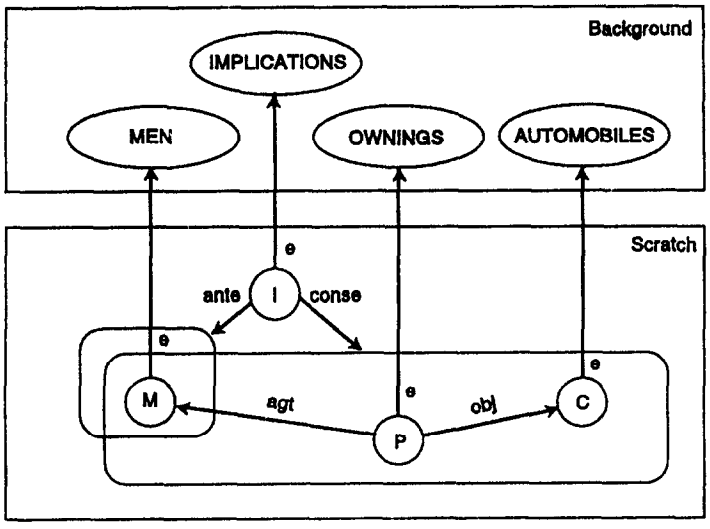


Figure 12. Hendrix's 'partitioned' semantic network meaning "Every man owns a car." Each e means INSTANCE-OF, **agt** and **obj** are the AGENT and OBJECT case-relations, **ante** and **conse** are the antecedent and consequence of the logical implication **I**, and **P** is the main predicate 'owns.' Anything in the **ante** space is considered to have the universal quantifier 'every': $\forall m(Men(m) \Rightarrow Men(m) \wedge Owningsc(P) \wedge Automobiles(c) \wedge P(m,c))$.

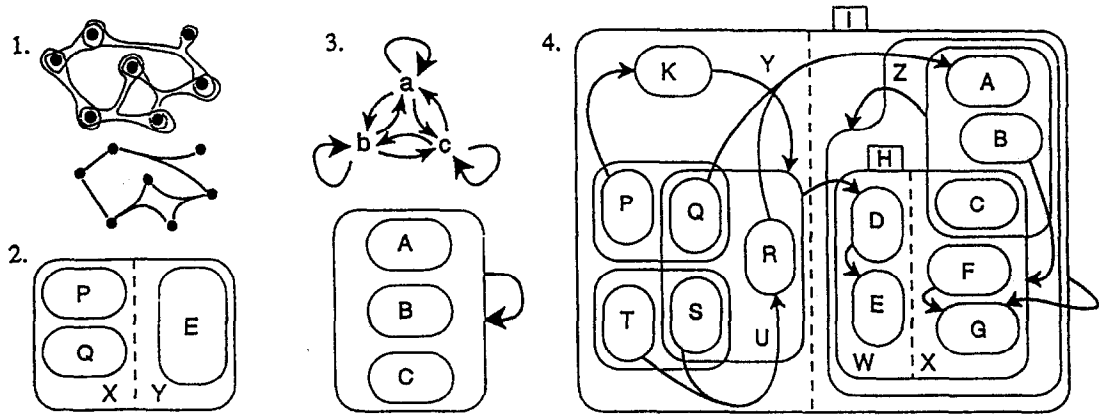


Figure 13. HiGraphs. 1. Two representations of a hypergraph with some 2- and 3-ended hyperarcs. 2. The Cartesian product notation, in this case $X \times Y$ with $(P \triangle Q) \subseteq X$ and $E \subseteq Y$ where $\triangle$ means exclusive-or. 3. The complete digraph on three vertices requires nine arrows; the HiGraph for the same system appears below it, needing only one arrow. 4. A HiGraph for a complicated system, including directed-hypergraph arrows.

Part 3 of the Figure shows how numerous relation arrows between individuals may be represented by a few arrows (in this example just one) joining blobs; the improvement is more dramatic the larger the graph is.[35]

CONCEPTUAL GRAPHS (see Sowa's summary) use a different system of *nested contexts* derived originally from the nested negations in Peirce's Existential Graphs. All connectives and quantifiers in logic can be handled by adding nested negation capsules to the existing implicit conjunction and existential quantification, so there's no need for special nodes for conjunction, disjunction, implication, equivalence or any other connectives, nor for a universal quantifier. All the logical connectives are definable with NOT and AND—among other possibilities—and $\forall x P(x)$ is definable as $\neg(\exists x \neg P(x))$. The negation capsules are nestable but cannot overlap. This is the

[35]Esch [129] points out that Conceptual Graphs diagrams may be combined with HiGraphs. HiGraph capsules may be drawn in the same plane around the concept nodes to represent class memberships, in order to augment or replace the usual 'type lattice' used in Conceptual Graphs.

most elegant treatment of logic in semantic nets.[36] In the following surveys of ECO, SNePS and PATH-BASED INHERITANCE, a net may have special nodes representing logical connectives or in some cases quantifiers. Automatic deduction using all of predicate logic is inherently intractable, so most KL-ONE systems have eschewed full first-order predicate logic in an attempt to make calculating subsumption tractable. Woods [130] describes various combinations of logical quantifiers in KL-ONE-style definitions.

A **vivid** knowledge base [131, 132] is one in which scoped negation, disjunction, universal quantification and functions are deliberately curtailed or eliminated. The knowledge base is a conjunction of true propositions containing existing objects and conjoined predicates and relations (this is just how the 'model' in model-theoretic semantics is described). Vividness was proposed by Levesque as a way to achieve tractability while retaining useful expressions and inference methods. Negatives are rarely used in ordinary life outside the specification of rules ("No smoking") or the defeat of a default expectation ("Cats with no tails"). There are too many negative facts—fully describing what's on a table may take a long time, but inventorying what's not on it will take forever. Also, *arbitrary* disjunctions like 'DOCTOR OR DEBENTURE' are almost never used, as opposed to approximate categories in a hierarchy like HEALTH-PROFESSIONAL = 'DOCTOR OR DENTIST' or interval tolerances like DIAMETER = 42mm ± 0.8mm.[37] Psychological studies show that people handle vivid information most easily [133]. A good vivid knowledge representation is simply the classic unextended semantic network.

Part III. World-Structuring Systems

"These ambiguities, redundancies and deficiencies recall those attributed by Dr. Franz Kuhn to a certain Chinese encyclopædia entitled Celestial Emporium of Benevolent Knowledge. *On those remote pages it is written that animals are divided into (a) those that belong to the Emperor, (b) embalmed ones, (c) those that are trained, (d) suckling pigs, (e) mermaids, (f) fabulous ones, (g) stray dogs, (h) those that are included in this classification, (i) those that tremble as if they were mad, (j) innumerable ones, (k) those drawn with a very fine camel's hair brush, (l) others, (m) those that have just broken a flower vase, (n) those that resemble flies at a distance. The Bibliographical Institute of Brussels also resorts to chaos ..."*

—Jorge Luis Borges, *The Analytical Language of John Wilkins* [134]

8. WANTED: ONTOLOGIES

The net representation of knowledge is only half of the story. The question remaining is: What to represent? The articles in this volume address both subjects and the interaction between them.

To solve a problem at the conceptual level, you must pick an **ontology** for your application area. In philosophy, ontology is the study of the concepts and categories of the world or, in Quine's words, What There Is.[38] In a particular AI application it is not always obvious what the categories, objects, attributes, entities and conceptual structures actually are. Many applications require both a specialized ontology and a general world ontology. IS-A hierarchies and thesauri (like *Roget's*) are ontologies. The semantic depth of the model of the domain depends on the richness of the ontology.

[36] The published definitions in Existential Graphs for *equivalence* and *exclusive-or*, however, rely on the use of coreferent propositional variables; ideally a pure Existential Graph shouldn't need these.

[37] Nontechnicians may well wonder why all this is a realization worth mentioning. Logicians' interest in deduction with the *modus ponens* inference rule made them preoccupied with implication and therefore disjunction since $A \Rightarrow B$ is more briefly represented as $\neg A \vee B$ than as $\neg(A \wedge \neg B)$. Frege [27] used implication itself as his primitive for this reason. Likewise, Peirce first used disjunction and universal quantification as the basis of his 'Entitative Graphs' until he saw the light and abandoned them in favor of Existential Graphs based on conjunction and existential quantification—he 'vivified' his semantic network in 1896 [24] but retained the nested negation loops for full expressiveness. He called such a representation 'iconic' because the net is like an icon or picture of the meaning in which the existing objects and relations are each directly represented.

[38] Some Husserlian ontologists have reinvented semantic networks! [135]

Most AI ontologies since the **Situation Calculus** was presented [136, 137] treat *situations* or *states of the world* (snapshots of the predicates which hold at a particular time) as objects which can participate in relations. A situation is changed by an *event* to a new situation. Such 'event-pulse' models have also been enhanced with more explicit representations of time [138]. Virtually every ontology has some notions of time and space.[39]

Common situations in ordinary life are described with stereotyped frame systems or 'scripts' which amount to specialized mini-ontologies. For example, there is a 'restaurant-script' network for a visit to a restaurant which describes the usual features like 'WAITER SERVES FOOD TO CUSTOMER' in their proper sequence. There are script-based systems for understanding natural-language reports in specific subject areas, as well as *Story Understanding Systems* which have script-like descriptions of standard plot themes [145].[40]

As the subject narrows further, modern technical thesauri for specialized fields are abundant ontological sources and some are immense [146]. They are used for automated document retrieval and the subject of computerized thesauri is now a solid part of modern Information Science. The article by Mili & Rada deals with the *Medical Subject Headings* (MeSH) classification with over 50,000 medical terms classified in 15 tree-structured hierarchies.

8.1. The Cyc *Project*

The most ambitious semantic network project at present is the *Cyc* project under the direction of Douglas Lenat and R. V. Guha [104, 126, 147]. Its main goal is to provide an ontology for the real world. This is a 10-year project to put in a knowledge base almost all of the factual knowledge that a person is assumed to have before reading an encyclopædia. This includes 'common-sense' knowledge, such as that people have physical existence, families, lifespans and desires, that two solid, rigid objects cannot occupy the same space at the same time, and so on.[41] Simple desk-encyclopædia knowledge will also be added. *Cyc*'s creators anticipate that there will eventually be around 100,000,000 axioms or facts in the knowledge base.

Including such a massive amount of knowledge is proposed in order to overcome the 'brittleness' of current expert systems, as illustrated in the broken gas pipe example in Section 1.2 in which the expert system was unaware of the ridiculousness of the IF-THEN rule because it had no underlying ontology for gas pipes, leaks, rock stars, or geometry. Instead of relying on suggestive names which lack effect on the system's behavior,[42] *Cyc* would have concepts and hierarchies and constraints for the semantic categories of gas pipe, rock star, etc. in a deep and complicated knowledge structure available for use by its inference algorithms. It is claimed that, unlike current expert systems, one based on *Cyc* could be substantially changed or redirected and the *Cyc* background knowledge would permit it to handle unexpected kinds of facts and rule interactions without falling apart.

Cyc uses a frame-based semantic network along with a modified logical language for constraints among slot values (such as a constraint that the age of your daughter must be less than your own). The semantic network apparatus is in the *CycL* language [126]; it is derived from Lenat's earlier work [124, 125] and is not part of any of the eight research families surveyed in this volume. Although it follows the 'scruffy' tradition of its predecessors (not worrying too much about formalization), *Cyc* now has a program which is supposed to maintain a mapping between *CycL* and the more formally respectable (logic-resembling) constraint language [52].

There are over 20 specialized algorithms for inference like *inheritance* (including the slot-inheritance mentioned in Section 6.3), *classification*, *'transfersThrough'* link composition

[39] A situational ontology with time and space is included in Hartley's article. The well-known *Situational Semantics* system of Barwise & Perry is presented in [139] (for a version using Conceptual Graphs, see [140]). Ontologies are included in various logic-based semantics systems for natural language like *Montague Semantics* [141] and the system of R. M. Martin [142]. Schubert & Hwang's *Episodic Logic* [143, 144] has an ontology of situations, events, beliefs, time, causation, etc. (See the ECO survey in this volume.)

[40] See Lytinen's CONCEPTUAL DEPENDENCY survey. Zarri's article "The Descriptive Component of a Hybrid Knowledge Representation System" describes a 'template-hierarchy' of frames which prescribe expected features of various ordinary events and human activities in the world.

[41] Such knowledge is presumably common to almost all adults, the *core* being even more widespread ('horse sense').

[42] This is a seductive trap in computer science, one into which *Cyc* itself occasionally falls [148].

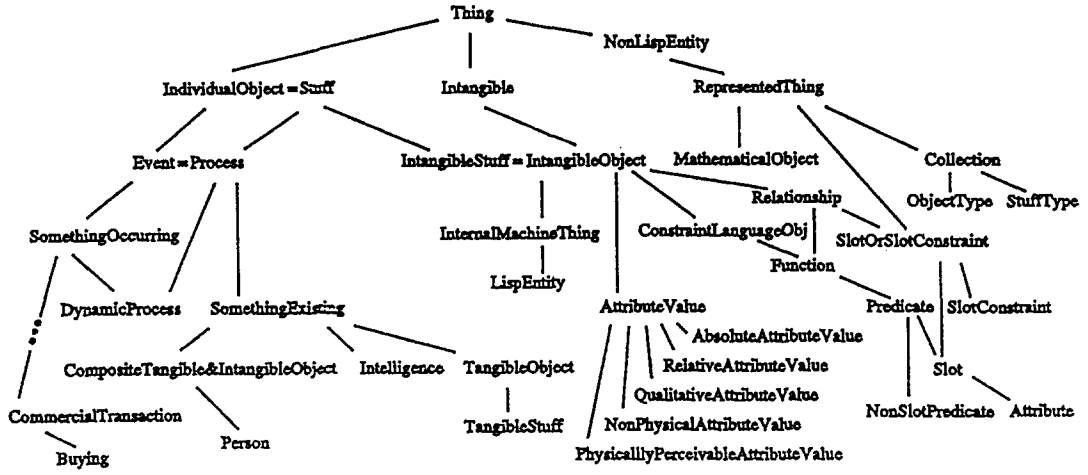


Figure 14. Part of the upper portion of *Cyc*'s main ontology.

(also described in Section 6.3), backward- and forward-chaining *rules* (implemented as attached IF-NEEDED and IF-ADDED procedures in slots; see Section 4), and *inverse linking* (e.g. supplying an *Over* slot pointing in the opposite direction of every *Under* slot), in addition to general automatic theorem-proving methods for logical formulæ in the constraint language.

Cyc is divided into **microtheories** which are ontologies for specialized subject domains. There are specific microtheories to deal with time, space, knowing agents, etc. Tigers, atoms, and the Rule Against Perpetuities would appear in different microtheories. *Cyc* workers are trying to provide interfaces between microtheories by which objects first described in one microtheory will automatically appear appropriately described in another, related one. The main microtheory **MEM** (Most Expressive Microtheory), for the world in general, is intended to represent 'consensus reality' or common knowledge. Figure 14 shows the upper reaches of the hierarchy.

This ontology has some surprises. A person, or any physically existing object, is classified as a *Process*, because it engages in existing during its lifetime. A substance or *Stuff* turns out to refer to the same things as *IndividualObject* because every piece of *Stuff* is an *IndividualObject* and vice-versa. However, the properties inherited via *SubstanceType* differ from those inherited via *ObjectType* in that only the former are preserved when an object is cut to pieces; the pieces of a wooden table are wooden but they are not tables. Each substance has a 'granularity level,' the small size at which it is no longer considered a substance but a collection of objects. For sand it is grains, for an element, atoms. These in turn are pieces of other substances so there is an alternation of Object-Substance-Object-Substance as the scale changes. *Event* and *Process* are the temporal versions of the spatial *IndividualObject* and *Stuff*. A predicate is given a default time of persistence; if Fred has a haircut at noon the system should assume his *HairLength* is the same (within a given tolerance) that evening.[43]

Defeasible reasoning is very important in *Cyc* since about 95 percent of its knowledge is defeasible (i.e. subject to exceptions). The usual problems of inheriting conflicting information abound. To resolve a conflict, competing *arguments* are analyzed; the methods of doing this are vague or scruffy at present but they have the merit that they are part of the knowledge base rather than a complicated revision of logic [112].[44]

Propositions can be reified (treated as things) and these things can be in minds. A sentient agent such as a human being, robot or higher animal is classified as a *CompositeTangibleIntangibleObject* having dual existences as *TangibleObject* (body) and *IntangibleObject* (representation, mind).[45] Collective agents like corporations or governments are provided for. All agents may have *Beliefs, Goals, Dreads, Purposes* and *Desires*. Case relations (roles) are *ActorSlots* in an *Event* frame.

[43]This addresses the 'frame problem' (nothing to do with 'frames') in AI: deciding which predicates change truth value as a result of an event and avoiding having to specify which ones *don't*.

[44]This is not to say that *Cyc*'s logic is pedestrian. It uses five truth-values: *True, DefaultTrue, Unknown, DefaultFalse* and *False*, with appropriate truth-tables for negation and the connectives.

[45]They may have different ages, to accomodate Dr. Frankenstein's monster.

8.2. The Holotheme

J.L. Jolley's **Holotheme** [149] also classifies everything, but in a single scheme based on structural complexity. Any object or relation is put in a broad pigeonhole-category represented as a string of bits in which four facet-dimensions (see Section 12.4) are substrings (each dimension's cardinality is a power of 2). The bit pattern for the basic categories is xxx.xxxx.xxx.x. Julius Cæsar or any real person, for example, is in basic category 101.1000.011.0.

The first dimension, **integrative levels** (3 bits), is based on overall generic complexity. For objects the chain is: members of sets (000.0), FULL SETS (000.1), points (001.0), GEOMETRIC FIGURES (001.1), photons (010.0), PARTICLES (010.1), atoms (011.0), MOLECULES (011.1), organelles (100.0), CELLS (100.1), organs (101.0), PLANTS & ANIMALS (101.1), departments (110.0), ORGANIZATIONS (110.1), local governments (111.0), NATIONS (111.1). The more independent units, in upper case, are indicated by the first bit of the second dimension, **formative grades**; its other three bits distinguish among *units*, *collections*, *series*, and *systems* of discrete *objects* at each level, or else *simples*, *mixtures*, *laminates*, and *runs* of *substances*. Often a full system built up from a lower level amounts to a unit at the next level up (as in *Cyc*). The third dimension, **semantic types** (3 bits), classifies active/passive attributive/entitive relations/terms, where each slash is a binary division; *qualities* are passive attributive terms (001) whereas *activities* are active entitative relations (110). The final bit for reality/fiction distinguishes Canada (111.1000.011.0) from Ruritania (111.1000.011.1). The meanings of the dimensions are not fully independent. A relation has bits (in reverse order) for: reflexive (aRa ?), symmetric ($aRb \Leftrightarrow bRa$?), transitive ($aRb \wedge bRc \Rightarrow aRc$?) and 'transversive' (is a's class unlike b's, as in $a \in b$?). Inclusion, for example, is 0101. Numerous mathematical notions are similarly classified.

Jolley notes that in this ontology 111.1111.111.1 refers not to God but to an interactive system of earthly sovereign states. The digestive tract is a *series* at the *organ* level. The Holotheme model as developed so far seems 'too neat' in aim while the actual classifying is 'too messy.' The usefulness in AI of such a scheme depends on the amount of information which can be inherited from a basic category due solely to its value in a particular dimension (along with any other information stored directly in the category).

8.3. The Wordtree

The Wordtree by Henry G. Burger [150] is a unique attempt to define word-meanings and concepts by combining pairs of more primitive concepts. It includes an immense vocabulary of what he calls transitives (like transitive verbs) each of which is a combination of two others. There is a base level of 44 primitive undefined verbs, each with a putative opposite, e.g. CREATE/UNCREATE, SPATIALIZE/VOID and CONVEX/CONCAVE, from which the rest are built using an operator '&' (which only sometimes means real logical conjunction).[46] To Fasten is to Hold & Stay, to Stay is to Unchange & Spatialize, To Nitch is to Fasten & Bundle, etc. See Figure 15. The book contains a valuable ontological model even though it is idiosyncratic and the author is more interested in the thoroughness of the word list than its logic or consistency. Its best use may be to inspire a more formalized 'concept-tree' adapting its ideas [151]. His word senses should be separated and the meanings of 'opposites' and the various senses of the '&'s should be semantically disambiguated. The *Wordtree* system depends on the non-equivalence of concepts $X = A\&(B\&C)$ and $Y = (A\&B)\&C$, i.e. the '&'s must not be associative.

8.4. Causal and Evidential Links

Many ontologies use **causal links** to form a deep causal model underlying the behavior of a physical system. Causal links are specialized relational links which indicate the propagation of change.

Causality involves dependency and time. In a mathematical formula like $P = VA$ each variable depends on the other two so a change in any one affects the others and there is no causal

[46] Burger suggests that 43 of the 44 may be further decomposed but omits the decomposition. His one primordial primitive is CREATE/UNCREATE.

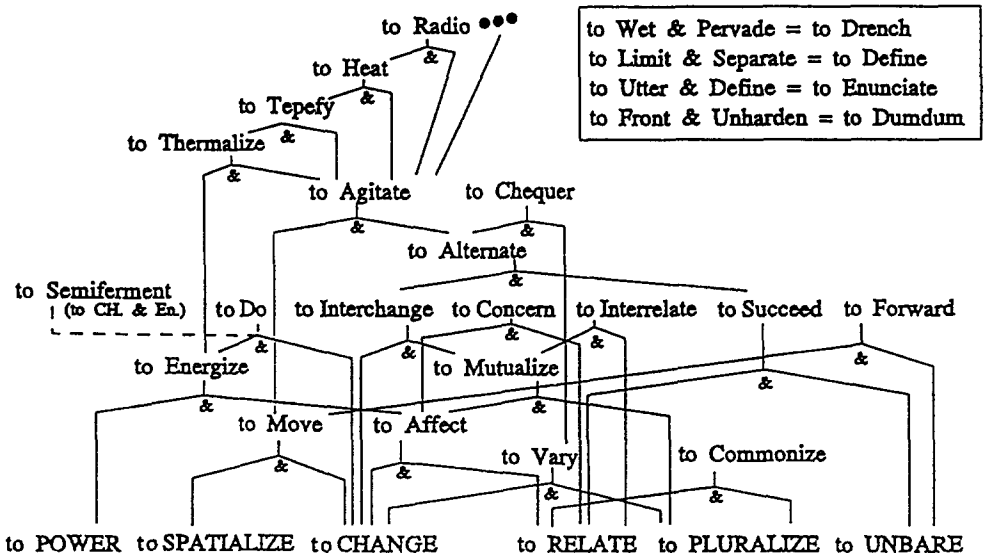


Figure 15. Examples of definitions in *The Wordtree*. The words in capital letters are six of the 88 primitives (including opposites), from which the rest are made.

ordering. In contrast, throwing a ball through a window *causes* the glass to break—breaking the glass does not cause a ball to have been thrown. Causal models of dynamic systems use influence diagrams to show which components influence which others; this may also show which parts of a system are irrelevant.

CONCEPTUAL DEPENDENCY (see Lytinen's survey) offered various causal links between states and events:

- *Event* **result** *State*—the event brings about the state.
- *State* **enable** *Event*—the state makes the event possible.
- *State* **disable** *Event*—the state makes the event impossible.
- *Event* **instrument** *Event*—the first event is instrumental to the second.
- (*Event* or *State*) **initiate** *MentalState*—the event or state brings about the mental state.
- *MentalState* **reason** *Event*—the mental state gave rise to the event.

Rieger & Grinberg [152] distinguish several species of causal dependency linking different combinations of actions, tendencies, and states, as shown in Figure 16, a causal diagram for a thermostat. The general causal direction is counterclockwise. In the *Confluence* link an output is controlled by a formula relating two inputs.

Similar influence diagrams in the AI field of Qualitative Physics [138, 153, 154] can be rendered in semantic networks as specialized concepts and relations, and the rules governing such diagrams (constraining equations based on numeric values or on qualitative predicates like ***hot, high, rising***, or ***exploding***) can be incorporated within the network itself or added to the inference procedures. A network of causal links can be *overlaid* on an otherwise non-causal semantic network.

Causal networks have been used in medical diagnostic programs as 'deep' knowledge underlying the observed correlations of symptoms [155]. For example, Patil's ABEL system [156] represents causal relations at several levels of detail, in which each higher level network is described in the more detailed network below it. Figure 17 shows three levels of relations among certain acid-base disorders of metabolism: first the highest and simplest clinical level of observed disorders, then an intermediate level, then the pathophysiologic level showing the most detail. Graph-theoretically, a net of causal arrows may be embedded in the more detailed net below it, and corresponding nodes are linked by vertical lines which show that certain conditions exist at multiple levels.[47] See also [103].

[47] It may be that *all* representations of the physical world or other complex domains should have multiple levels of description like this. See the article "Granularity Hierarchies" by McCalla, Greer, Pospisil & Barrie.

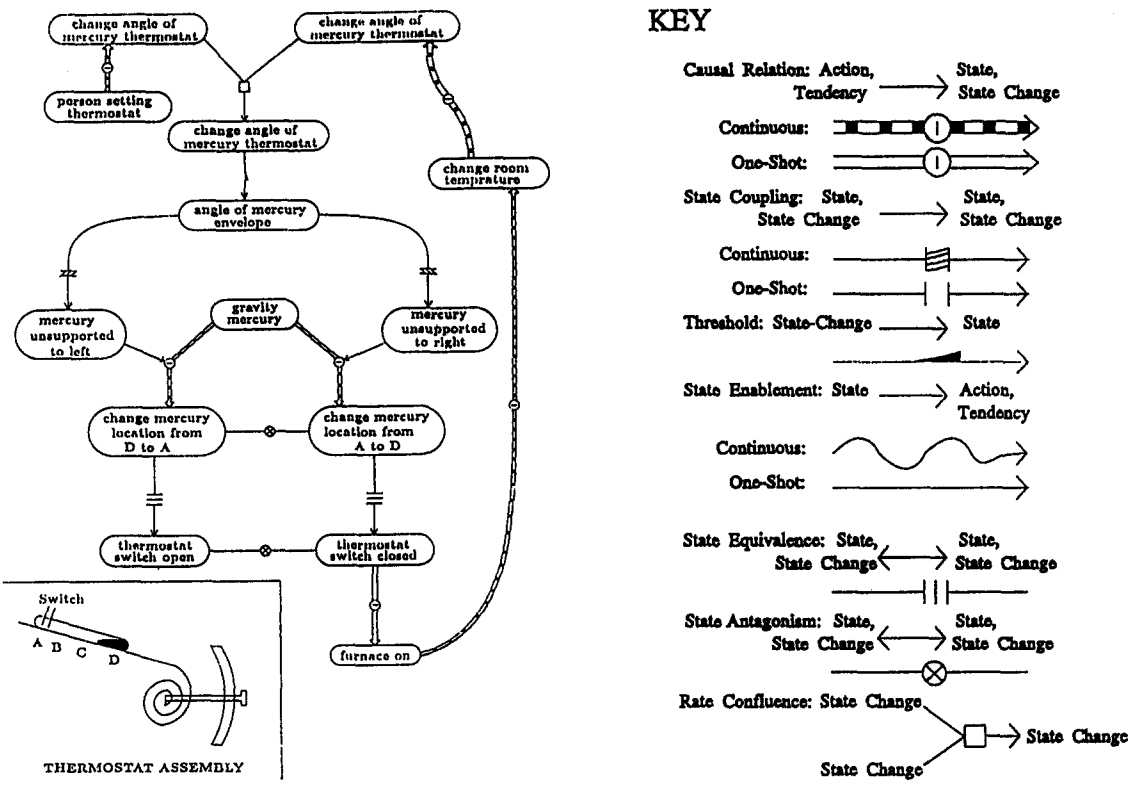


Figure 16. A causal network for a thermostat in Rieger & Grinberg's net notation. Each arrow is a specific kind of causal influence.

Every directed causal link between two states or facts involves two **evidential links** (in opposite directions) between them. Although fire causes smoke and *not* vice-versa, fire is evidence of smoke, and smoke is evidence of fire. Pearl [107] uses a probability-based network of proposition-nodes and causal links (see Figure 18) with numeric weights to indicate how strongly one proposition-node supports (is evidence for believing) another. The probability of any one node being true depends on all the other nodes in the network, except that certain nodes become *irrelevant* if blocked by certain sets of supporting nodes. Finding these *separating sets* in his system depends on causal direction. In Figure 18, "The sprinkler was on" is good reason to believe "The grass is wet" and, separately, "The grass is wet" is good reason to believe that "It rained last night," but "The sprinkler was on" is certainly no reason to believe "It rained last night." The belief path from P_2 to P_1 is said to be "blocked" by Q. However, due to the converging causal directions of the links, P_1 and P_2 are not separated into independent sets by knowing the truth of Q. Because both P_1 and P_2 are known only as possible *causes* of wet grass, they become mutually dependent—knowing either one destroys the support for believing the other. (There's no such dependence or blocking among *effects* R_1 and R_2.) Pearl calls this dependence and blocking effect the *defining characteristic* of causal direction and claims that it doesn't depend on time's direction.[48]

Causality is controversial, although most people agree at least that something in the cause must happen earlier in time than something in the effect [157]. A true causal link involves both an actual physical link (or trajectory or force) between things and an evidential dependency between associated propositions.

[48]He also has a probabilistic interpretation of multiple, defeasible inheritance using this kind of weighted evidential link as a defeasible IS-A link. "Birds fly" means that if all you know is that Tweety is a bird, your expected probability that Tweety does not fly is below a very small threshold called *epsilon*, but this may be revised if you gain new, relevant information (e.g. Tweety is a penguin) that renders the Tweety→BIRD link inoperative [107].

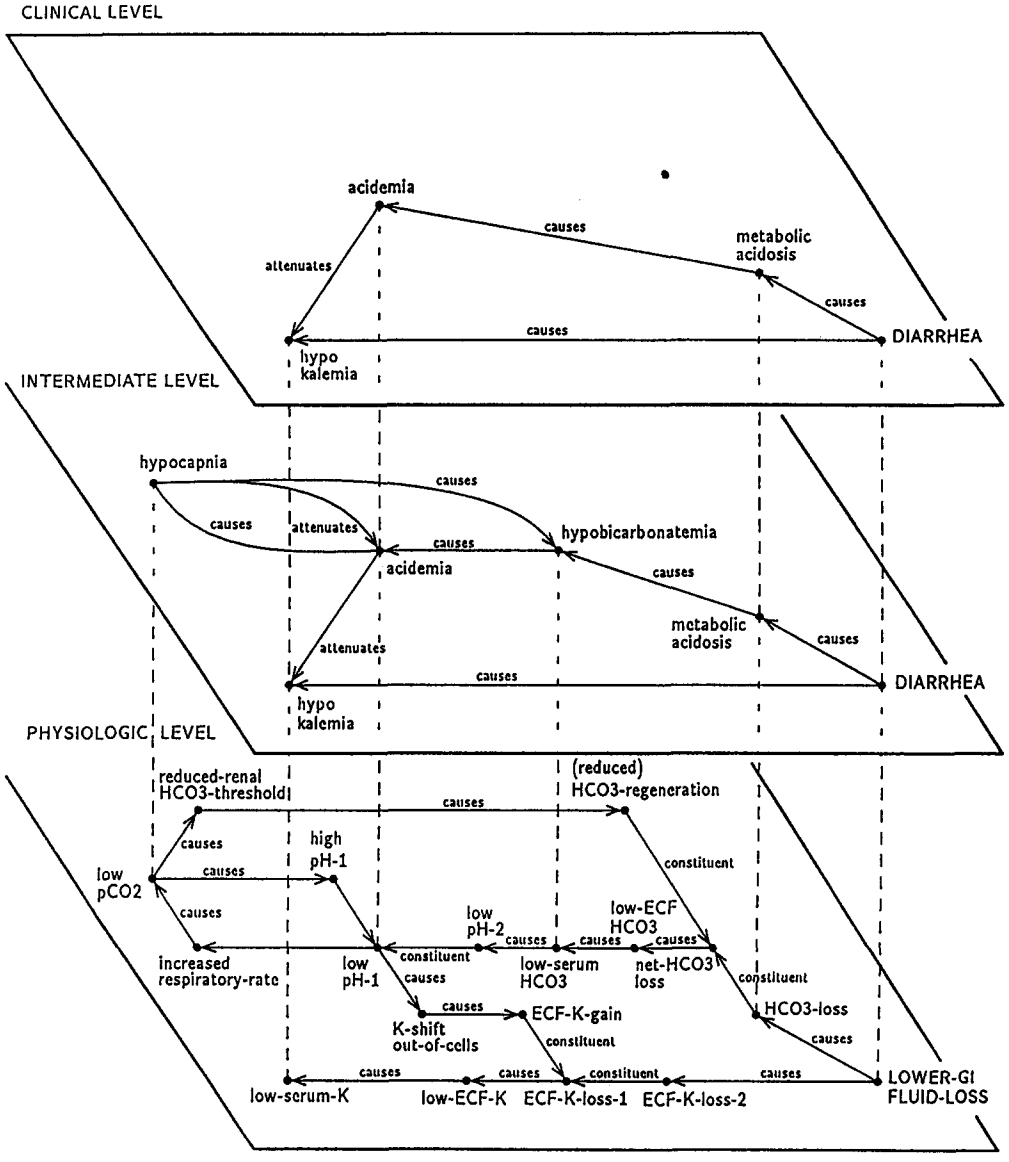


Figure 17. A causal network showing three levels of increasing detail. This is Patil's model of electrolyte disorders with diarrhœa as the cause, shown at all three levels. Certain conditions exist at multiple levels, indicated by the vertical lines.

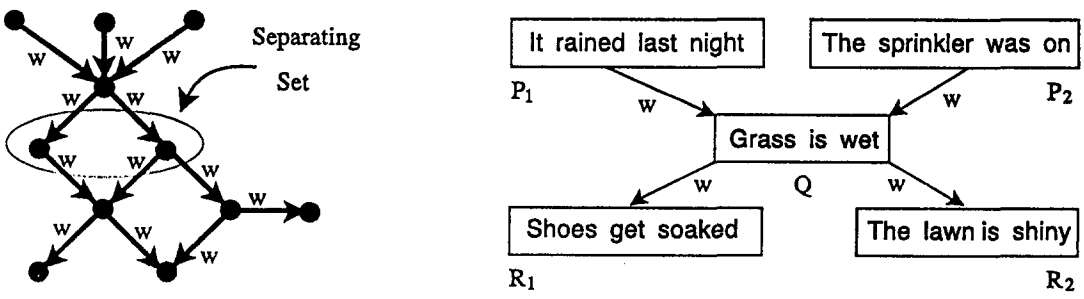


Figure 18. Pearl's causal network. Left: A network with a 'separating set' of nodes based on the causal arrowheads; knowing truth values of nodes in this set makes the upper part irrelevant to the lower part. The 'w's stand for the probabilistic link-weights indicating degree of support. Right: If Q is known it supports all of the other nodes, but if P_2 becomes known, Q ceases to support P_1. The arrowheads show the causal direction of otherwise bidirectional *support* links.

8.5. *Semiotic Links*

Another specialized kind of link is the **semiotic link** between a representation and that which is represented. There is such a link between the actual physical pattern of the tracks in the London Underground (subway) and the the well-known schematic map of the system. There is another such link between the map and my mental recollection of the map. The pattern 'travels' along these semiotic links from the represented thing to my mind. Most information in the original pattern is lost in the process but certain information is preserved. This is equally true of more abstract conceptual patterns like semantic nets.[49] A semantic network which represents entities which may themselves represent (such as minds, texts, maps, etc.) needs semiotic links.

9. DISCRETE NETS IN A CONTINUOUS WORLD

A semantic network is a discrete structure as is any linguistic description. Representation of the continuous 'outside world' with such a structure is necessarily incomplete, and requires decisions as to which information is kept and which is lost.

In pattern recognition, a transition is made from continuous input data to a schematic spatial representation (by means of low level signal processing and 'feature extraction' of features like object boundaries) and thence to a symbolic description. Semantic nets describing the scene or situation are generated only at the end of this process. However, a net describing possible expected objects may be used to help disambiguate ambiguous input, for example in semantics-driven parsing programs which eliminate semantically nonsensical interpretations. If there is an expected arrangement of certain objects and spatial relations then, after initial preprocessing, *Syntactic Pattern Recognition* can parse a two- or three-dimensional scene into semantically meaningful objects based on a formal *graph-grammar* [159, 160]. (See also the article "Introduction to Graph-Grammars with Applications to Semantic Networks" by Ehrig, Habel & Kreowski.)

Often, reasoning is done with a dual representation: both a continuous mathematical model (e.g. a spatial representation, map or system of equations) and a discrete semantic network or linguistic description. Geographic and pictorial databases [161, 162] require such an approach.[50] CAD systems normally require too much precision to be useful for interpreting linguistic spatial descriptions; if I say "Put on your hat" I don't specify X and Y coordinates. Some systems are being developed which allow looser, natural-language-like descriptions using case relations (prepositions) which 'underspecify' position [165–167].[51]

9.1. *Semantic Control of Continuous Models*

Dynamic physical systems from thermostats to oil refineries to aircraft in flight are modelled in *control theory* which uses differential equations to describe how the outputs of a system vary depending on the inputs, and yields corresponding 'control laws' for setting the inputs so as to keep outputs within a desired range. In real-time operation a controller supplies control signals based on the values of system sensors; the control laws are fixed in advance. In complicated environments with unexpected events and changing missions and priorities (such as air combat) classical control theory is inadequate since new models and new control laws must be selected from time to time based on semantic *descriptions* of the model states. This is called **semantic control** and, because of their multiple levels of abstraction, semantic networks are the language of choice for state description [169].

[49] Marty's Peircean article "Foliated Semantic Networks" treats such links as an extension of the dyadic *'concepts,'* linking individuals and their attributes, described in Wille's article. Peirce himself, however, insisted that semiotic links are inherently *triadic* sign-relations since a representation can only represent an object *to* someone. Zadrożny is formulating networks of triadic sign-links [158]. IS-A links may ultimately be triadic, with the *purpose* of the classification as the usually omitted third argument. *IS-A(porpoise,mammal,biology)* but *IS-A(porpoise,fish,fishing)* in some locales.

[50] The article by Rucker & Aldowaisan proposes two multilevel representations, of language and of vision, with a 'juxtaposition' of high-level spatial case relations at the proper places in a high-level schematic diagram. Similar approaches are taken in [163, 164].

[51] Picture analysis based on eliciting spatial prepositional relations from line drawings is described in [168].

Semantic control theory adds *rules of change* for the concepts, features and values in a classical Systems Theory state description, to form what amounts to a network of transitions in time between different semantic nets. The rules trigger and constrain state transitions. One semantic net represents the current *semantic state* and others represent goal states; the system must explore a space of state changes from one net to another according to the rules of change and the latest information, since the nets also change in real time [170]. When a path to a goal state is found, a series of 'program generators' generate the necessary executable functions for the controller which correspond to the path of desired state-changes [171]. The change rules may dictate discrete changes in the net like addition or deletion of objects or relations—the system is therefore self-modifying—and these changes can be specified by an augmented graph-grammar. This is a natural development of Mesarović's "teleological self-organization" extensions to Systems Theory [172].

Semantic control involves using AI techniques to reason at a higher level of abstraction than the base level of system sensors and controls. The hierarchies in semantic nets allow this; instead of dealing with wing flap controls, an aircraft control system may deal with high-level strategies like *"sacrifice enough weapons to extend flight range, avoid ground radar #407, distract enemy MiG #3 for ten minutes and head for the nearest landing spot."*[52] Hierarchic, multilevel description is used in semantic control in order to ascend to the highest applicable level of abstraction and perform all possible reasoning at that level. Using the result, the system selects or perhaps automatically designs the control model, and finally creates the control inputs in accordance with classical control theory. The full armory of AI techniques may be used at the higher levels within the limits of real-time processing.

10. RELEVANCE AND SALIENCE

It is important in reasoning to consider the *relevant* objects, features and relations in a situation and to disregard the obviously irrelevant. Requiring a causal connection, as described above, is one approach. A severe problem of knowledge bases based on ordinary logic or default logic is that all of the facts in the represented world have a chance to bear upon any question. To accept a default inference a system must inspect the entire knowledge base just to insure that no other fact will implicitly override the inference—a huge waste of time.

10.1. Limiting Access

Semantic nets, because they are directly implemented as structures in a computer, offer an automatic indexing or arrangement of knowledge in which some knowledge is more easily accessible than the rest. A common presumption is that facts relevant to a concept will be reachable from that concept by following link paths, and that relevance will diminish with distance in the net. Some systems, like KRL [56] and some of *Cyc*'s specialized inference components (Section 8.1), simply cut off access during query processing at, say, 25 links away from the queried concept.

Another way to limit the search to relevant nodes is to require similarity in some respects. Cluster-based **associative networks** use numerically weighted similarity-links.[53] For example, Findler [178] defines a cluster of nodes relevant to any given node by calculating weighted association links between the nodes, based on an average of 'closeness' in several feature dimensions (like *size, severity, age*, etc.), including in the relevant cluster only those nodes linked with a weight over a certain threshold. The cluster structure is thus derived from the values of node

[52] Various abstraction hierarchies occur in knowledge-based simulation and control systems, some based on a description of physical structure, others on a description of behavior (functional task decomposition); see [173] which uses both. STEAMER [174] has multiple levels of *object* abstraction and HIRES [175, 176] uses levels of *process* abstraction. For interacting parts of a machine, Joskowicz [177] presents a hierarchy of physical object assemblies' kinematic behavior dependent partly on internal structure (using an abstracted 'configuration space' of the parts' possible configurations given their shapes and constrained freedom of movement) and partly on the purpose of the assembly. Straightforward PART-OF hierarchies are also used.

[53] True *semantic* networks have specific meaningful relational links.

attributes and very similar nodes are bound to be in the same cluster. The SHRIF system [179] based on this method is a currently implemented medical information retrieval system which wastes no time on searching irrelevant areas of the database.

To achieve tractability in searching for relevant facts, the search space in a semantic net may be compartmentalized. **Access-Limited Logic** [180, 181] is a formalized frame-and-rule-based semantic network system in which any information not accessible via a chain of relations (links) cannot be used in a rule. Also, every slot-value pair or rule is in one or more overlapping *'partitions'* (subsets of the entire net); in automatically answering a query, a sequence of rules may not cross the boundaries of a partition. There is no global access to rules. The knowledge base as a whole is only 'Socratically complete' which means that only by a getting the right series of explicit questions (which each allow crossing of a partition boundary) can the system be guaranteed to make every valid inference using the whole network. Rule-based inference is thus effectively compartmentalized (exhaustive search occurs only within a partition) except when the user chooses to look farther afield. The partitions are to be "somehow semantically cohesive."

Subject area taxonomies called **Topic Hierarchies** may be used to limit search to relevant areas. They are not quite the same as the generalization hierarchies for objects and events discussed so far. The concept CHICKEN has many links to others, but the *relevance* of these depends on the topic; if the topic is Embryology certain links of CHICKEN are relevant but not the SEASONINGS link to TARRAGON. Although it is little realized, a Topic Hierarchy is ultimately based on some (genuine) generalization hierarchy of people's purposes. If your purpose is to cook a chicken then the embryological, historical, or evolutionary information is largely irrelevant. Within a military topic there may be several alternative descriptions of the same terrain, the relevance depending on the task at hand.[54] Topic Hierarchies for controlling access to semantic networks are explained in the ECO survey by Cercone *et al.*

The AI and Cognitive Science fields increasingly view **analogy** as a central part of everyday reasoning. Analogy establishes parallels between features of different objects, events or situations, and may be based on a mapping between the graph-theoretic structures of the two applicable semantic nets [182–184]. Metaphors like "Richard Nixon is the submarine of foreign policy" are intended to transfer only *salient* features from submarines to Nixon: lethality, hidden movement and surprise attacks perhaps, but not that he takes to water or has propellers in his rear. To limit the relevant features available for analogy, Way [185] uses a system of 'salience masks' to mask out irrelevant parts of the generalization hierarchy.[55] The question "Why is a raven like a writing-desk?" is annoying because the only notable common generalization is PHYSICAL-THING and no feature is salient.

11. DICTIONARIES AND SEMANTIC PRIMITIVES

If nodes and links are to be more than meaningless tokens there needs to be some way to probe their meanings. One approach is to integrate a formal **lexicon** or **dictionary** closely with the database of assertions. Each definition in the lexicon is itself a semantic network which is accessible to the computer. There have been extensive efforts to use existing natural language dictionaries as the source for a formal conceptual lexicon for AI [17, 186].[56]

Quillian's *Semantic Memory Model* [187, 188], the best known of the early semantic networks in AI, is an encoding of a dictionary. Each word-sense has a *plane* which includes the network representing its definition. See Figure 19. Labelled links are used within a plane including *superclass* (i.e. IS-A), *modification*, *subject/object*,[57] and logical links *AND* and *OR*. A typical

[54] Even within strictly 'scientific' topics there are variations. There are different affinities among biological species depending on whether genotype, phenotype or surmised evolutionary history is the primary taxonomic criterion.

[55] Weber's connectionist machine for handling salience-based metaphor (using special *aspect hub* nodes) is described in Shastri's article in this volume.

[56] Slator's article "Sense and Preference" describes one such effort based on *Longman's Dictionary of Contemporary English*, which uses a restricted set of words in definitions and has codes for various categories of word meanings.

[57] Other case relations like *from* are not links but are treated as separate verb-like words with their own subject/object links. The double arrows from USE in Figure 19 are subject and object links.

definition specifies a superclass and modifications of it, thus implementing inheritance. Each word is defined at a 'type-node' for the word which points to a defining plane for each of the word's senses; all uses of the word (in other definitions) are 'tokens' (pointers) which point to its type-node.

In processing natural language text it is necessary to pick the most likely among multiple senses of a word, that is, to disambiguate. The **semantic intersection** between two concepts is the set of concepts which best relate the two, found by *spreading activation* in which links are followed and nodes activated in breadth-first fashion starting at both source concepts. Where the two spreading 'spheres' of activation intersect, there is the result. Thus CRY and COMFORT intersect at SAD as shown in Figure 19; this is used in interpreting text to show which of the various senses of crying and comforting are related.[58]

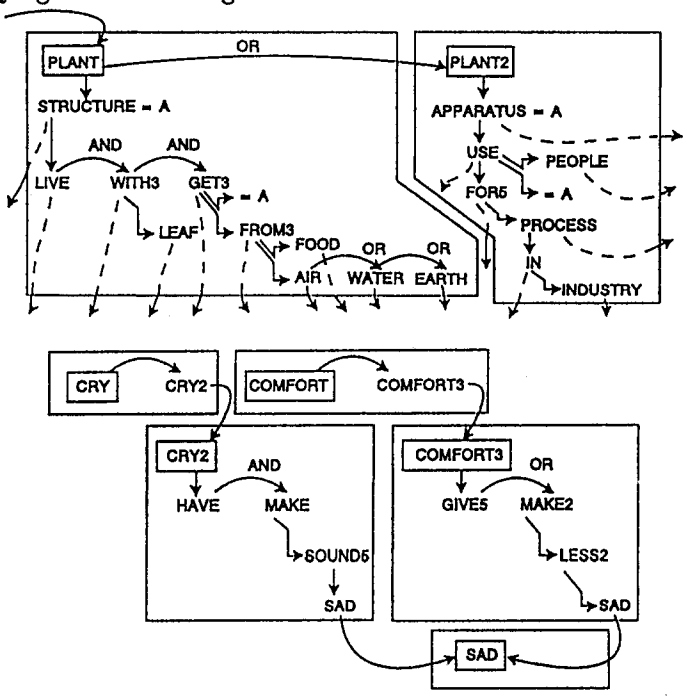


Figure 19. Quillian's representation of word senses. Top: Two senses of the word "plant." A "plant" can be a structure which lives, has a leaf and gets its food from air, water or earth, OR an apparatus people use for a process in industry. The dashed links are pointers to other defining planes. Bottom: The semantic intersection of CRY and COMFORT is SAD—one sense of CRY means MAKE a SAD SOUND, and one sense of COMFORT means MAKE LESS SAD. This helps choose the preferred senses of "cry" and "comfort" in a text containing both.

Quillian considered the *full meaning* of a word to be the sum total of all the structure built by recursively substituting for each word its definition. In a semantic network, substituting the defining net of a concept for the node representing the concept itself is, in fact, a *graph-grammar substitution* of a complex graph for a simple one; see Figure 20. Such a substitution is an increase in granularity and detail, a semantic 'zooming in.'[59] If a question answering system looks up a concept-dictionary definition and the structure of the defining network is not adequate to decide the question, infinite regress seems possible as each defining term is itself looked up, as in the case of natural language. Either there is some elemental level of conceptual *primitives* which are undefined, or there are *vicious cycles* (concepts defined in terms of each other) in the definitions.

Whether true **semantic primitives** exist is controversial [190, 191]. The best-known primitives proposed for AI are those in Schank's original CONCEPTUAL DEPENDENCY theory (see

[58]Hendler's article "Massively Parallel Marker-passing in Semantic Networks" describes special computers which do this directly in hardware for disambiguating word senses in text.

[59]Much like net refinement, the *hypernet refinement* of [189], or the expansion of descriptive *construct graphs* [122]. See also the article on formal graph grammars by Ehrig *et al.*

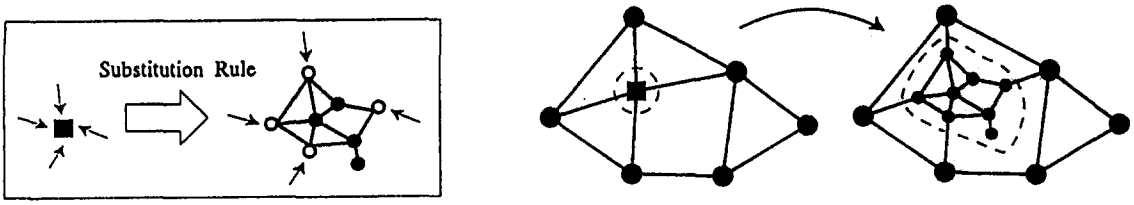


Figure 20. A graph-grammar substitution rule is used to expand a node into its defining net. The small arrows indicate the set of attaching points which must be preserved.

Lytinen's survey). The notion of semantic primitives is not new—it was the basis of Leibniz' *Ars Combinatoria*[60] and Bishop John Wilkins' splendid *Essay towards a Real Character and a Philosophical Language* [196] as well as many of the other 17th Century attempts at an ideal philosophical language or *Characteristica Universalis* for inter-cultural communication [37, 134, 197–199].[61] Axiomatic systems such as Whitehead & Russell's *Principia Mathematica* [206][62] or Hayes' Naïve Physics programme in AI [207] (or the proposed interplanetary language *LINCOS* [208]), are continuations of those attempts, as are, in essence, semantic network ontologies like *Cyc*.

Wierzbicka claims in *Lingua Mentalis* [209] that "All sentences in all natural languages can be paraphrased in terms of these thirteen signs ..."

> *I, you, someone, something, world, this, want, not want,*
> *think of, say, imagine, be a part of, become.*

and, using them, she painstakingly paraphrases numerous verbs at length in [210]. As noted above in Section 8.3, *The Wordtree* uses 44 primitive terms along with 'opposites,' from *CREATE/UNCREATE* to *EMOTIONALIZE/DEEMOTIONALIZE*. Julius Laffal's *Concept Dictionary of English* [211], with 117 'conceptual primitives' like AGGR (aggression), COLR (color), FOND (loving,cherishing) and VAPR (vapors, gasses, mist), is used to annotate and interpret the babblings of psychotics (and the Declaration of Independence). The PREFERENCE SEMANTICS approach to text understanding (see the survey by Wilks & Fass) uses about 100, but does not claim they are the ultimate primitives. Igor Mel'čuk's massive ECD (Explanatory Combinatory Dictionary) project uses 53 primitive relations [212].

Litkowski seeks semantic primitives using a graph-theoretic model of an existing dictionary [213]. Each word-sense is a node, each use of it in the definition of another, a directed arc (as in Quillian's model). The resulting digraph for the dictionary is immense. A graph-reduction algorithm reduces it to a core of mutually defined primitive concepts which contains the smallest possible set of 'circular definitions.' All other concepts are defined using concepts in the core. Instead of primitive undefined *concepts*, there are only primitive *cyclic structures*. A first phase of this technique reduced a set of 20,000 verbs in *Webster's Third New International Dictionary* to under 4,000 [214, 215].

Some complain of the inefficiency of a complete reduction to primitives (granting that it is possible at all) due to the considerable cost of processing and storing fully expanded descriptions,[63]

[60] Inspired partly by the attempted combinatory philosophy of Ramon Llull [192, 193] and partly by the ingenious conceptual combinations in written Chinese characters, this system uses prime numbers to represent primitive concepts and combines them by multiplication to create compound concepts, based on the unique factorization of any number into its constituent primes [194]. An elaboration which combines the constituents and preserves the *order* in which they are given, called Gödel-numbering [195], is based on multiplying powers of consecutive primes.

[61] Compositional 'ontological grammars' built from primitives occur in the systems of Lodwick (or Lodowyck) [200], Dalgarno [201] and Wilkins, as well as the carefully devised languages of two then-reported central Australian civilizations (one consisting entirely of hermaphrodites) [198]. Pasigraphic languages (compositional systems of primitive 'self-evident' symbols or hieroglyphs) are also relevant, like [202] or the more recent *SAFO* [203] and *Semantography/Blissymbolics* [204, 205]. Original editions of many works of this type are in the *Philip Mills Arnold Semeiology Collection*, Rare Book Department, Olin Library, Washington University, St. Louis, Mo. 63130.

[62] The later volumes—evidently unread by anyone, Russell said—include a primitive-based ontology of number, continuity, distance, change, etc.

[63] In ordinary serial computers, that is. Parallel lattice operations are actually improved; see Section 13.2.

but a system formally *based* on primitives need not *use* them in the normal course of business. It is only when the finer-grained definitions become relevant that recourse need be had to the dictionary. This is a matter of reasoning at the right level of detail or granularity, which depends on the task at hand.

12. INTERSECTING INFERENCE HIERARCHIES

The hierarchy of the world's concepts is not arbitrary; we do not use or need a mental concept for every possible combination of qualities, relations or objects. We have a relatively sparse and efficient encoding of the structure of the world, partly inborn and partly learned, based on economy of mental effort and subjective usefulness. The world's customary categorical system has a divided structure in that there are major components which are mutually independent or only interact in constrained ways. The conceptual structure of trust law, for example, has few connections to that of bee anatomy. Various shape descriptions may be interrelated but they are all independent of color. A high priority for this field is to 'factor' the conceptual hierarchy or 'tease out' component sub-hierarchies which are amenable to easy computation. In other words, Divide And Conquer. This may seem like the recent AI trend towards hybrid systems of independent, specialized reasoning components, but I believe all the components can operate within the unified mathematical framework of Order Theory (the theory of ordered sets including Lattice Theory), as discussed below.

An assertional semantic net (relational graph) participates in many different partial orderings which may be used for rapid inference; these are derived from the meanings of its nodes and links as well as from its pure graph structure.

12.1. Inferential Semantics

One result of the original work of the Cambridge Language Research Unit (see Section 2.2) is *Inferential Semantics* by the late A.F. Parker-Rhodes, a difficult and largely unappreciated book belatedly published in 1978 [216]. It combines mathematical Lattice Theory with semantic, syntactic and even tonal analysis of sentences. Two kinds of lattice are used: a lattice arrangement of the semantic relations underlying a sentence, called a **Rhema Graph** (like the assertional relational graphs discussed earlier), and separate inferential lattices of what he calls the '**base domains**' of **labels** on the nodes in the Rhema Graph.[64] This elegantly separates assertional links from inferential links.

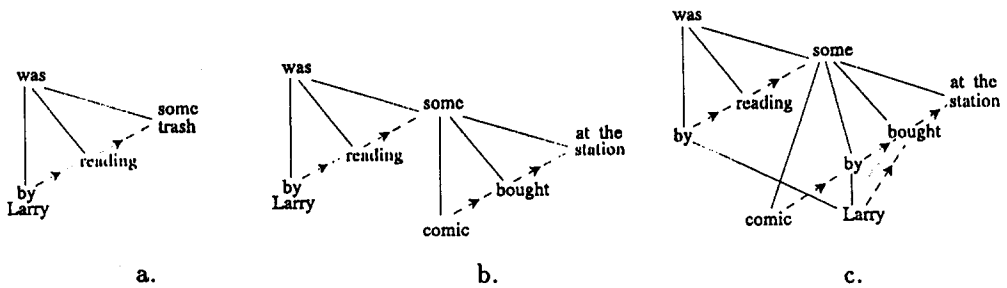


Figure 21. Rhema Graphs based on the sentences: a. "Larry was reading some **trash**." b. "Larry was reading a comic bought at the **station**." c. "Larry was reading a **comic** he'd bought at the **station**." (Boldface type indicates emphasis. The various node-labels for the nodes' different inferential dimensions are omitted.)

To make a Rhema Graph (Figure 21), Parker-Rhodes converts a syntactic parse tree (sentence diagram) of a sentence into a partially ordered set (Directed Acyclic Graph) by fusing any leaf-nodes which refer to the same thing. He then converts it to a complete lattice by ordering any dangling branches using a **focus-ordering** (from 'focus of attention'). The focus order

[64] A third kind, a lattice of natural language *syntax*, was proposed in 1961 [217].

distinguishes between a point of reference and a point of interest.[65] His Rhema Graph diagrams 'take some getting used to.' In Figure 21 the solid lines represent the main semantic relations and the dashed arrows represent the focus ordering. Several Rhema Graphs are combined into one giant one for an extended discourse or text.

The *inferential* lattices apply to a Rhema Graph in the following fashion. There are several *labels* on each node in the Rhema Graph, by which the node participates in several independent external inferential lattices (the *base domains* of the labels) which operate in concert on the Rhema Graph. These are subsumption lattices in which any label has a position in its 'base domain' lattice such that it is subsumed by all node-labels higher in that lattice and it subsumes all lower node-labels.

These base-domain lattices are more structurally advanced than those in other AI systems. There are different lattices for different aspects of meaning, and a node in a Rhema Graph may have labels from several of these meaning aspects or 'inferential dimensions.' Aside from the usual IS-A hierarchy or lexical lattice for objects, there is a quantifier-lattice, a case-lattice, an 'identification'-lattice, and others. Lattice Theory determines subsumption relations between any two (multi-labelled) nodes, and these relations in turn control the subsumption relation between any two Rhema Graphs.

He also builds the lattice of a particular base-domain from simpler *constituent* lattices. Simple structures are combined by lattice operations to create larger ones; the ultimate result is an abstract super-lattice which can be used directly by the computer for fast deduction (this super-lattice is only a 'virtual object' which need not actually be built or represented explicitly in memory).

As an example, consider the **case lattice** of 'deep case' relations (Section 5.1) like "to," "from," "along" etc. His cases are not mere tokens, but are related by three different ontological sub-lattices (constituents of the case lattice), namely *Phase, Mode*, and *Grade*, shown in Figure 22. *Phase* refers to directional orientation: 'source,' 'goal,' 'position' (where something is), and 'passage' (where it is not, what is passed by), all in *a*. *Mode* is whether the relation is passive or active: the two-element lattice in *b*. where states subsume events. *Grade* involves causal status, a three-way distinction among 'primary' (independent or spontaneous cause), "secondary" (causal result), and "tertiary" (stationary background involvement), represented by the lattice in *c*.

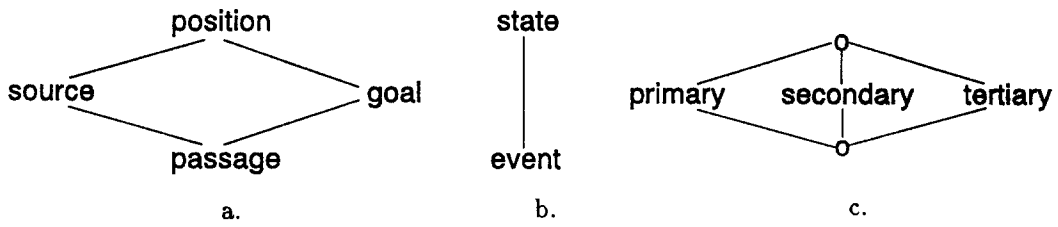


Figure 22. The lattices of the three sub-categories of the Case lattice: a. *Phase*, b. *Mode*, c. *Grade*. See text. The links represent subsumption in the lattices, e.g. any source is also a position.

The actual deep cases appear in the 'product' of these three lattices, the large lattice in Figure 23. There are more distinctions among these 25 cases than in the other case systems. **Donor** for example differs from **Receiver** in the obvious sense of directional opposite (difference of *Phase*), and it differs from **Sponsor** which is a static concept (difference of *Mode*), and it further differs from **Source** because a donor actively instigates a transfer, unlike a mere source (difference of *Grade*).

Another example, the elaborate **quantifier lattice**, includes, in addition to the quantifiers $\forall$ and $\exists$ of ordinary logic, the natural numbers (finite and infinite), at-mosts and at-leasts, plurals, ranges, most-of, set-ofs, fractions, etc. With this lattice as a factor in the inferential super-lattice

[65] It may depend on intonation: "Jane got **marr**ied" has a different focus-order meaning from "**Jane** got married" even though both sentences have the same logical representation. For human beings focus order seems to divide an assertion into an addressing part, which selects a memory address by the points of reference, and a storing part, which stores the points of interest there.

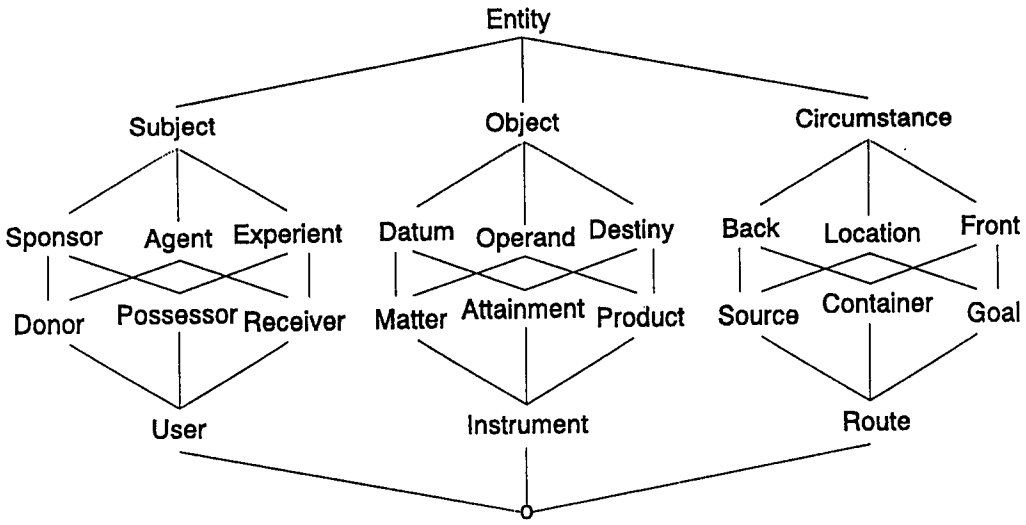


Figure 23. The Case lattice. This is the product of the three lattices in the prior Figure. Each element in this lattice has a meaning as a 'deep case' relation except the bottom (empty) case.

it can be automatic that a woman with at least four daughters is necessarily a person with children.[66]

A case-label and a quantifier-label would both label a particular node in a Rhema Graph, giving that node a place in both subsumption lattices.[67]

12.2. Other Order Intersections

Recent research has rediscovered Parker-Rhodes' approach, establishing lattices of subsumption relations between what are really labelled relational graphs, based on external partial orderings in which the labels participate. In his thesis, *A Lattice Theoretic Approach to Computation Based on a Calculus of Partially Ordered Type Structures* [219], Aït-Kaci presents a useful summary of Lattice Theory followed by his own approach to subsumption using ***abstract data types***. Data types in computer science are ***schemas*** which specify the arrangement and content of data structures. (For example, the ***complex number*** type is specified as a two-part structure consisting of a *real* for the real-number part and another *real* for the imaginary part.) Complicated data types are defined using nested record structures which we can recognize immediately as frame systems, like Kay's 'feature structures.'[68] Nested data types have an inherent tree structure (similar to the tree structure of macro expansions in computer languages).

Aït-Kaci converts these trees to nets by fusing certain nodes, using co-referential variables as 'tags.'[69] The subsumption lattice of these net-structured types depends on an external partial

[66]For another lattice of quantifiers, see Doudna's quantifier model [218].

[67]This approach could organize the hierarchies in other systems which often have meaningful compositional structures of lattice-factors. Schubert and his colleagues have explored factoring the type hierarchy into specialized dimensions, such as for time-intervals, color, part-relations, etc. (see the ECO survey by Cercone, Goebel, de Haan & Schaeffer), but without a lattice-theoretic foundation.

[68]Kay [220–222] introduced these for processing natural language. A feature structure is a nested frame with type restrictions on slot values, referred to earlier as a *schema*, which describes a constituent of speech, object, or event. Slots are labelled and referred to symbolically and there is no fixed number or order of slots. If the nested frames are represented as a graph, as in Figure 6 in Section 4 on Frames, they form a tree. Tagging certain slots with *coreference variables* marks those slots as being identical, i.e. they are fused. This converts the structure from a tree into a general digraph (branches may re-converge). This kind of structure is used in unification grammars for natural language like the one in Lytinen's article "A Unification-based, Integrated Natural Language Processing System" in this volume.

[69]This method of joining what would otherwise be branches of a tree is quite common. Boley does it at the implementation level in his article on hypergraphs, and Sowa's CONCEPTUAL GRAPHS approximate Peirce's original "lines of identity" (see also Roberts' article) the same way. Bound variables sprinkled within a linear string representation of a net are reminiscent of conventional logic's symbolic detritus, but I haven't found a satisfactory

```
X0:Prince(SON-OF => X1:Queen(MARRIED-TO => X2:Consort;
                             RULES => Kingdom);
          ADMIRES => X2;
          MARRIED-TO => Princess(OBEYS => X1;
                                 BUYS => X3:Dresses;
                                 WEARS => X3;
                                 MARRIED-TO => X0) )
```

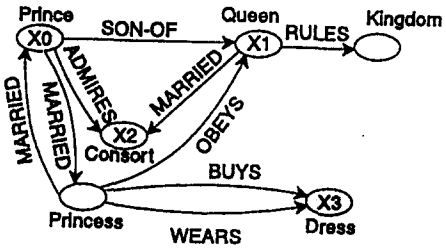


Figure 24. A sample type structure in Aït-Kaci's notation, with the graph representation on the right. The parentheses show the nesting. The X's are *tags* or coreference variables to show identity. Note that you cannot say in this system that a Prince is the SON-OF a Consort, since only one instance of each labelled outgoing link is allowed and the Prince is already a SON-OF the Queen.

ordering on simple terms (like the subsumption order in Parker-Rhodes' 'base-domain' lattices of node labels) represented by some arbitrary '$\leq$' relation. The criteria for whether one type subsumes another are partly fundamental (the subsuming net must be a subgraph of the other and all corresponding nodes in the two nets must obey the '$\leq$' relation) and partly artifacts of the chosen linear representation.[70] [71]

If relations have their own inferential hierarchy, as described in Section 6.3, then this too will partly determine whether one relational graph subsumes another. Corresponding *links* in the two graphs will have to obey an external '$\leq$' ordering for relations. This relational poset, the poset(s) of concepts, and the basic poset of graph inclusion (for directed graphs) must all be combined to determine the final inferential poset structure.

12.3. Determinables and Attributes

Frame systems, semantic nets and databases all use the object-attribute-value triad rather freely. The customary implementations have been criticized [229] for a lack of 'semantic relativity' among object, attribute and value. Formally the three arguments of a triad $R(x, y, z)$ are equal and undistinguished. What is it about one of the three that allows us to say that it is the attribute? Philosophers discuss attributes and their values as having the **determinable-determinate** relation [185, 230–232] such as the determinable 'color' having the determinate value 'red.' Some mistakenly think a determinate is simply a (more specific) *subtype* of the determinable, such as thinking that 'red' is a subtype of 'color'; it isn't. A determinable may superficially resemble a quality but it is really a higher-order predicate which specifies a named *range* of *interrelated* determinate qualities (values), one or more of which an object may have.[72]

In fact a determinable has the effect of splitting off a piece of the full abstraction hierarchy and permitting separate (and with luck tractable) computation within that piece. A determinable's range may or may not be ordered and the values may range over partial orders and lattices rather than linear orders. The label base-domains mentioned in Section 12.1 are examples of lattice-ordered determinables. See also the analyses of scale systems and attributes in [233–235]

alternative.

[70] There is a lattice-theoretic foundation. For automatic deduction by means of unification-based resolution theorem-proving, unification à la PROLOG can be done with these type-nets since the poset of types is 'completed' to form a true (Brouwerian) lattice with unique Greatest Lower Bounds as unifiers; see [222–224]. This corresponds to the unifiers called 'maximal joins' in CONCEPTUAL GRAPHS. Related German work on sorted logics and the inference order of formal type-structures appears in [225].

[71] The slots in a data type (or in the similar *'feature terms'* of [224, 226, 227]) must be distinctly named and have unique values; thus they are *functions*. Graphically, this means that all the relation-links leaving a certain node must have distinct labels (Figure 24). This restriction does not apply in semantic nets like KL-ONE in which a concept may have several instances of the same role, such as when a person's SON role is filled by many sons. This has a drastic effect on the formal complexity of the algorithm used to caculate subsumption of two graphs: with functional slots it is almost linear [219] whereas with multiple roles (or, equivalently, functional roles with *sets* as values) it is believed to be formally undecidable [106, 228].

[72] *Subranges* may be subtypes, however. "Crimson" *is* a subtype of "red" i.e. it is a more specific quality.

and part 2 of Wille's article in this volume. In music theory, for example, the four determinables *timbre, loudness, time* and *pitch* each have a different order structure [236].

12.4. Faceted Classification

Practical real-world taxonomies occur in library book classification schemes [237]. The familiar ones, Dewey Decimal and Library of Congress (as well as the Bliss system), are trees of subject categories. Instead of a simple tree arrangement, Ranganathan pioneered the idea of **faceted classification** [238–241] in the Indian COLON classification.[73] Each book has a main subject code followed by five **facets**: *Personality, Matter, Energy, Space*, and *Time*. See Figure 25. Each facet uses a separate tree-structured hierarchy of categories. Your desired subject "18th Century Japanese cast lead statutes" might have the general form **ART: Statuary - Lead - Casting - Japan - 18thC.**

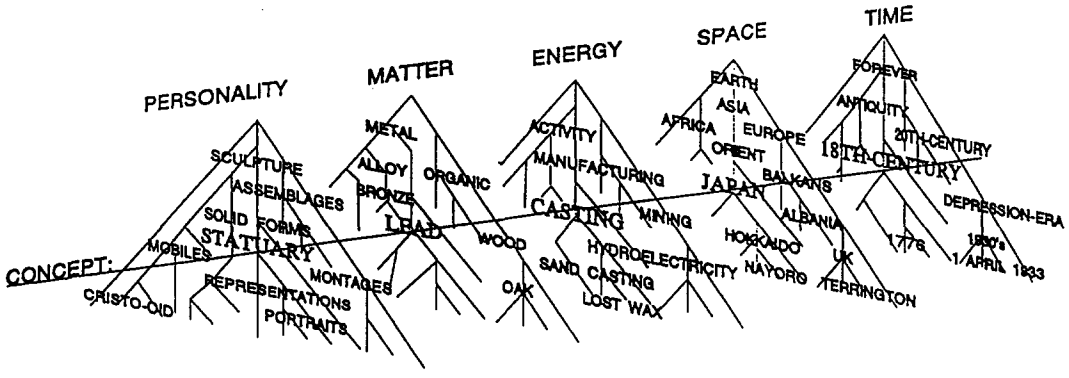


Figure 25. Separate tree structures for each facet impinge on "18th Century Japanese cast lead statues."

You might accept a book with broader subject matter, say "Antique Metal Sculpture of the Orient"; this would subsume your specific subject in each of the nonempty facets, so it covers your subject.[74] 'Faceted search strategies' derived from this work are now used widely in *Information Storage and Retrieval* in library and information science [146, 242]. Multiple inheritance exists only at the intersection of the facet trees and within each facet there is only single inheritance. Sometimes in AI the different trees are called *perspectives* and an individual is allowed to be an INSTANCE-OF classes in more than one tree although each tree allows only single IS-A inheritance from classes higher in the tree [243]. If this limitation is acceptable then there are ways to assign compact numerical codes to every possible subject such that determining subsumption for computer retrieval is very fast. The section on 'Type Hierarchies' in the ECO survey by Cercone *et al.* describes such a coding method. Senyk *et al.* [103] use a set of intersecting facet-dimensions in a medical database. Separate hierarchies (not all trees) for *Anatomy, Physiology & Pathophysiology, Clinical Findings*, and *Etiology (causes)* are used together to induce a *'Nosology'* hierarchy of diseases with specific relation-links to the values in the other hierarchies. As noted earlier, there is no fundamental reason that a facet (determinable) could not have any arbitrary ordering rather than just a tree ordering.

To what extent useful knowledge can in fact be separated into fairly independent facets or determinables is a crucial question, discussed further in the conclusion.

[73] Some limited faceting is also used in the Universal Decimal Classification.

[74] The actual COLON codes are complicated. To arrange books on a linear bookshelf, the code system has to yield a single tree structure. It is obtained by jumping back and forth among the facet-trees during the subdivision process as separately prescribed for each main subject based on practical considerations. This is unnecessary in computerized Information Retrieval since you can access a book or article based on all facets at once. A code can be quite specific: "The Prevention of the Side Effect of the Antibiotic Muscular Injection to a Cow with Mouth Disease in the Rainy Season of 1971" is **KX,311,21;423:6,66;63;4:5.4=3'N71-u**. The breakdown is: Animal Husbandry. Cow. Mouth. Virus Disease. Treatment. Antibiotic. Muscular injection. Side-effect. Prevention. Tropical Asia. Rainy season of 1971.

13. FAST ANSWERS

Since a knowledge-based system needs to *infer* answers beyond retrieving the explicitly stored information, query processing is generally slower than in simple database systems. On the other hand, classical semantic networks provide faster answers than is possible using (generally worst-case intractable) automatic theorem-proving methods based on logic. Various graph-search techniques can be exploited. If the formal expressiveness is less than logic's then the worst-case complexity of restricted forms of inference can be tractable with a trade-off between what kinds of things can be said and how fast automatic inference can theoretically be [106, 131]. Even with the full power of logic added, a semantic network still permits those speedy inference mechanisms which can make use of its structure, which are also among the most common and direct inferences that people make. Formal complexity results have been published in most of the main research families but this has been the special concern of KL-ONE, especially the worst-case time complexity of computing terminological subsumption between defined concepts (Section 6). An earlier exhortation by Levesque [131] that this must be doable in polynomial time is now being rebelled against in favor of retaining maximum expressiveness, as discussed in Woods & Schmolze's KL-ONE survey.

The most common method of automatic deduction in logic-based programs (like PROLOG), 'unification-based resolution,' can be speeded up dramatically using an *order-sorted logic.* This assigns every logical variable to a particular *sort* or 'type' or class of individuals. The types are arranged in an IS-A hierarchy used for fast unification. The first half of Cohn's article "Completing Sort Hierarchies" explains this further.

13.1. Databases

Practical semantic network systems require large databases for storing asserted nets along with concept hierarchies and/or a concept-dictionary or lexicon. The usual access methods for retrieving records may be used to retrieve frames in frame-based semantic networks. In Information Retrieval, semantic nets are needed for accurate retrieval of relevant documents in preference to using keywords or sets of features (so a search for cases of dogs biting men won't retrieve "MAN BITES DOG"); the document *subject descriptor* is in semantic net form although the actual text is not [244].

For a factual database, an important issue is the speed of finding assertional nets. Usually, during automatic question-answering, a **query graph** is compared with the existing relational graphs in the database. Fully general searches for graphs, or subgraph embedding algorithms, are worst-case intractable but regularities and constraints in semantic nets allow fast average-case access.[75]

Database theory has been split between network-structured databases and relational databases. The now-dominant relational database model identifies a record or object entirely by the values of its attributes; no explicit links between records or classes of records are available to the user. This eases the conceptual burden on users of large and fairly homogeneous business databases, but it is too unstructured for users of complicated multilevel databases with interrelated definitions of very different objects, as in Computer Aided Design. A recent trend is to model databases on semantic networks in order to make use of the structure of the concepts and relations in a data model. Chen's *Entity-Relationship* database model [246] and the *Nijssen Information Analysis Model (NIAM)* [247, 248] are obviously close to being semantic nets. Approaches such as Sandewall's *Information Management Theory* [249, 250] apply semantic net *schemas* to underlying relation-tuples as constraints and type-restrictions; see also [251]. The AI and database fields are gradually melding in the form of Knowledge Bases [252–254] and Objected-Oriented

[75] This applies to direct searches for query graphs, as described in Levinson's article "Pattern Associativity and the Retrieval of Semantic Networks," as well as to enhanced searches which yield related graphs, generalizations or specializations based on the abstraction hierarchy. Ellis [245] takes direct advantage of the abstraction hierarchy in efficiently storing and retrieving Conceptual Graphs; the actual graphs are not stored, rather the *differences* between them in the hierarchy are stored using a graph-grammar formalism. Both systems use the patterns of previously stored graphs to guide storage of a new graph, gradually building up a partially ordered set of graphs.

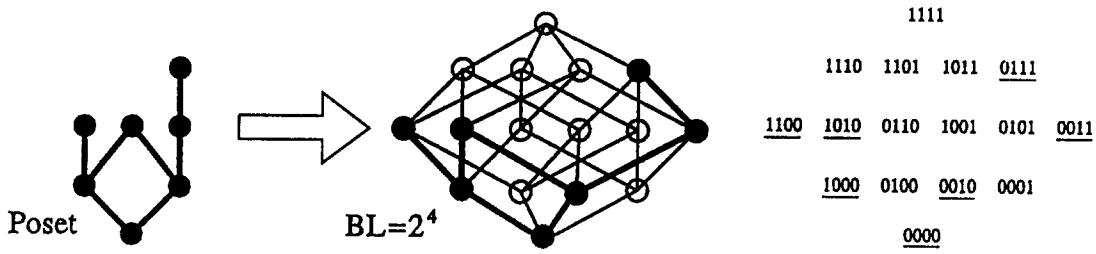


Figure 26. Plunging a poset into a Boolean lattice BL (in this case 2^4 represented by a four-dimensional hypercube). The nodes in the poset are assigned the corresponding bit-codes (shown underlined) which can be used for one-machine-instruction lattice calculations.

Databases [255, 256]. An example is the TAXIS system described in the PSN survey by Mylopoulos.

13.2. Precompiled Lattice Codes

Concept hierarchy operations like inheritance, classification, unification, etc. depend on calculating subsumption of one concept by another or the common specialization of two concepts. For tree structures, the numerical codes described for *Types Access* in the ECO survey by Cercone *et al.* may be pre-calculated for every node in the tree; subsumption testing then becomes a simple matter of a numerical comparison. For the general case, very fast calculation (in a few machine cycles) is possible if the hierarchy structure is a *Boolean lattice* of 2^n nodes in which every node is represented by a distinct string of n bits, by using parallel bit-wise logic operations on the bit-strings.[76]

For example, the string resulting from a single AND of two such bit-strings represents the common specialization of the two concepts, and subsumption can be determined in one more operation (since $a \leq b \Leftrightarrow (a \wedge b = a)$ in any lattice).[77] The only problems are (a) the limited word-length in the computer's CPU (the number of bits acted upon in one operation), and (b) the effort of converting your existing hierarchy structure into a Boolean lattice in advance while preserving the hierarchy relationships.

The first limitation, due to word length, only slightly degrades ideal performance and could be cured by using a long-word-length machine. The second problem of creating an appropriate lattice is solved by embedding or plunging your existing hierarchy into a Boolean lattice, as illustrated in Figure 26.[78]

If only you could use a Boolean lattice in the first place! If every concept in the world were a certain combination of some of n totally independent primitive (and meaningful) qualities, each concept in the lattice would be represented by an n-bit code with each '1' meaning that a particular quality applies; this is maximum efficiency since only n meaningful, primitive bits are needed to represent the 2^n concepts. Such a 'flat' set of qualities is inadequate to capture the rich structures in semantic network hierarchies, but the closer you can get to using such primitives as

[76] It is called a 'Boolean lattice' because Boolean logic operations AND, OR and NOT (or set operations INTERSECTION, UNION and COMPLEMENT) are the lattice operations *meet*, *join* and *complement* in this lattice. A diagram of a Boolean lattice is drawn as a square, cube or n-dimensional hypercube, upended on one point, as in Figure 26. It has 2^n nodes where n is a number of independent binary-valued variables (bits); the nodes represent every possible combination of those n values. Formally, a **Boolean lattice** is defined as a distributive, uniquely complemented lattice.

[77] In bit-wise logic, each bit position in a word is operated upon independently and in parallel. The AND yields the Greatest Lower Bound, or *meet*: 1011 AND 0110 = 0010, where 1 means 'true' and 0 means 'false.'

[78] Aït-Kaci *et al.* [257] describe a procedure for doing this plunging which takes advantage of typical patterns in hierarchy structures and results in a compact lattice. Polyakov & Dunaev's use of the smallest possible set of their 'meronomic words' to distinguish among taxonomic categories amounts to another compact Boolean embedding [93, 94]. Cohn's article "Completing Sort Hierarchies" in this volume describes an alternative 'Boolean completion' method designed to accomodate arbitrary unions and intersections in addition to the poset of concepts supplied by the user.

building-blocks of concepts the fewer the extra bits needed to encode the Boolean lattice. This is an *efficiency* argument *for* semantic primitives.

14. SEMANTIC NETS IN PARALLEL HARDWARE

Specialized semantic network machines have been designed in which the network pattern is embedded directly in computer hardware consisting of many processors linked together physically. The processors acting simultaneously (i.e. in parallel) can take advantage of the parallel structure of a net. In some designs each node in the net has its own processor. Fahlman's NETL language [19] was created to allow processing of inheritance hierarchies by a specially designed parallel machine; this is described in the survey by Thomason. Hendler's article describes other *massively parallel marker-passing machines* for semantic networks and analyzes the use of the *Connection Machine* for this purpose—a well-known parallel computer with over 64,000 interconnected processors operating simultaneously. Three other *structured connectionist* computers for semantic networks are described in Shastri's article (called "structured" to distinguish them from current primarily homogeneous neural nets).

Despite a superficial resemblance to semantic nets, *neural nets* often take the opposite approach to Artificial Intelligence. The most typical neural nets, inspired in part by proposed models of biological neurons, are networks of interconnected simple processing units which have numeric weights on the connections. Information is not necessarily supplied in any symbolic form at all—the net learns from repeated examples of input patterns and adjusts the weights on all the connections in a 'training phase', after which the net is able to discriminate among new patterns and recognize those for which it is trained. No node or link has any specific meaning; any meaning or information is spread over the whole pattern of weights. Many enthusiasts exult in the lack of discernable symbolic representation and savor the self-organizing, bottom-up, non-analytic aspect (the 'black-boxers'). Others such as [258–263] seek a synthesis of neural nets with symbolic, analytic representation of knowledge and reasoning (as in semantic nets); see the citations at the end of Shastri's article. The key aspect is the *compositionality* of the internal representation; in semantic nets, as in most thought, we can analyze what goes into an idea and how it is structured—this is the essence of semantics. The conceptual gap between neural 'subsymbolic' processing and intelligible symbolic compositional structures like semantic nets is only now being explored [264].

Hinton [265] stores the concept-relation-concept 'triples' of a semantic net associatively in a neural net (the net will fill in any one part of a triple given the other two). Each concept has, in addition, a separate set of meaningful 'microfeatures' or numeric values representing primitive qualities. The network treatment of the microfeatures accomplishes a limited form of inheritance as described above for a Boolean lattice of 'flat' qualities. Derthick's μ-KLONE [118] uses similar microfeatures to calculate subsumption among concepts and among relations, and also for 'role-filler-type-restrictions' which restrict the classes of individuals allowed to participate in a relation. There can be constraints among the microfeatures (e.g. some qualities are made mutually disjoint). In answering a query, this connectionist realization of KL-ONE, because of the weighted link implementation, *ranks* the stored descriptions by degree of consistency with the query-description, and answers with the *closest* available answer rather than the usual firm yes or no.

15. HOW TO IMPLEMENT A SEMANTIC NETWORK

In a conventional computer, semantic networks may be *custom-programmed*, implemented using built-in features of *Object-Oriented Languages*, or implemented in *Expert System Shells with Object-Oriented extensions*. Almost all commercial systems are frame-based. Some commercial *hypertext* systems with typed links and node hierarchies may be used to store information as

a semantic network, but these systems have depended on human traversal of of links (as in browsing) without built-in machine inference capability [266].[79]

15.1. Programming Semantic Networks

The most common way to implement a semantic network is to program it yourself. You have the freedom to tailor it to your needs with of course the corresponding hard labor and intellectual burden. A research-quality program for your favorite semantic network system may be available from its inventor. Most semantic network systems are developed in LISP, but Prolog, C and Pascal [13] have also been used. AI languages such as LISP and Prolog are better for developing and rearranging a semantic net program, while conventional compiled languages like Ada, C and Pascal, lacking their built-in evaluation functions, are faster for using the net program once its algorithms are stabilized. Many LISP textbooks describe semantic networks (see the end of Section 4), and in [267] semantic networks are the centerpiece of a clear exposition of logic programming.

15.2. Object-Oriented Languages

Object-Oriented languages, systems and databases include ideas already familiar in semantic networks and frame systems, like *inheritance* and *procedural attachment* (the procedures or programs in an object are called 'methods'). Objects (basically frames) communicate by sending each other *messages* (on arrival a message selects a 'method,' by name, from within the object)—*everything* is done by these named procedures in objects, and access is only possible through an object's message handler, i.e. objects are *encapsulated* and their internal structure is otherwise hidden. Inheritance is a fundamental feature of Object-Oriented systems and languages [268, 269]. Objects inherit attribute slots and values as well as 'methods' from classes and superclasses.[80] Object-oriented systems are mostly concerned with operations (such as inheritance) acting on the world of *data structures*, and languages like *Smalltalk* or *Objective C* have elaborate built-in ontologies describing their own data structures and operations. Object-oriented systems do not address the general goal of representing all kinds of meaning, so they are often deemed to be quite far from the field of AI. Nevertheless, the semantic network community should pay close attention to the developments in object-oriented systems because the speed and efficiency standards are high and the research is very active.

Given an existing object-oriented language it is easy to create a semantic network *using* that language. Object-oriented LISP systems like *CLOS* for Common LISP make this fairly painless.[81] Earlier semantic nets have been implemented in the object-oriented LISP extensions *LOOPS* and *Flavors*; the public domain *XLISP* [272] has such an extension. CONCEPTUAL GRAPHS have been implemented in *Smalltalk* [184]. Of course the quandaries that beset semantic networks, such as multiple, defeasible inheritance conflicts, are also troublesome in object-oriented systems.

15.3. Expert Systems with Objects

Many commercial Expert System Shells have optional or integral *object-oriented extensions* handy for implementing semantic networks. Typically a frame-based object/class hierarchy is combined with a rule-based expert system using a heuristic-search-based inference engine. *KEE, Nexpert Object, ART-IM, KnowledgeCraft, KES, Laser, Level 5 Object, GoldWorksII, AICorp-KBMS, Kappa, Arity/Expert* and others (often reviewed in *AI Expert* magazine and elsewhere) provide these facilities in integrated packages which include graphic interfaces, hooks to conventional programming languages and databases, etc., and are available on a wide range of conventional computers from microcomputers to mainframes. They allow you to browse around directly

[79] A hypertext is a network of linked text passages with machine-followable cross-reference links. A word in a displayed text may be selected by the user; this causes another text, pertaining (and linked) to the word, to appear. This new text may have further links to other texts.

[80] There are more kinds of object-oriented inheritance structures than just abstraction hierarchies; for example, *method combination* or *code sharing* of program fragments involves other kinds of inheritance [270, 271].

[81] *CLOS* allows you to handle multiple inheritance by specifying an arbitrary *ad hoc* preference-ordering among the parent superclasses of each class, to determine which parent 'wins' in a conflict.

in a screen diagram of the IS-A hierarchy, using a 'mouse.' Although they make no use of recent inheritance theory, these systems typically provide a repertoire of inheritance methods which may be specified for a slot, like an *override parents* or *union of parents' values* inheritance-facet for a particular slot. These systems are expensive and may seem slow, mainly because such a system has to support numerous features which may have nothing to do with your intended use of semantic networks.

16. CONCLUSION

The articles in this collection go further than I have in applying diverse disciplines to semantic networks and vice-versa. This rich and ambitious approach is to be expected for a practical computing method whose subject is 'the world.'

Anthropomorphically, I've referred to a computer's 'knowledge.' Can a computer 'know' *anything*? It is often said that computers merely mimic knowledgeable behavior based on arbitrary symbols which, to the computer, are without meaning since the computer has no direct experience of what they represent. I would answer that abstract graph-structure is an exception to this; a computer truly *has* internal graphs existing within it and can comport its behavior based on them. A semantic-network-based computer 'experiences' the abstract graph-structure directly, at least as directly as a person can, in the sense that internal graph structures result from sensory input and influence behavior systematically. A computer's experience of graph structure is not *faked*.[82]

Some disparage nets as a mere notational variant of symbolic logic (First-Order Predicate Calculus). Israel & Brachman [273] and Sowa [274] have made spirited defenses of nets from different points of view (model-theoretic and linguistic). According to Schubert [275], the difference between logic and semantic networks (broadly defined, as I do) is one of emphasis rather than fundamentals. Strict nets extended as in Section 7 and (higher-order) Predicate Calculus (with equality) are alternative notations for an underlying logic; the question is, do you care about the graph-theoretic structure of the relations and abstractions? If you can use that structure for inference, you do.[83] In logic a linear conjunction of many propositions normally shares numerous variable-letters which are free in any one proposition but are treated as bound 'place-holders' in the conjoined list taken as a whole. A description in Predicate Calculus is a 'ripped apart' semantic net which is, depending on the particular manner of the ripping, sprinkled with artificial nonce variables tagging where the ripped pieces fit together. This was noted by Peirce, who invented both.

Symbolic logic deals primarily with bricks and mortar, semantic nets more with principles of architecture. The semantic constraints of the upper levels of Figure 5 preclude a logical "jumbled heap of bricks." As noted in Section 7, nets favor 'vivid' knowledge over arbitrary logical constructs, even when the formal expressiveness is the same. Some cognitive linguists reject logic entirely as a basis for the meaning of language. Yorick Wilks (see his PREFERENCE SEMANTICS summary) says that, for one sentence, multiple possible meanings (semantic nets) contend with each other, and that various measures are used to pick the 'best' meaning among them—this is wholly alien to conventional logic. Of course the *ontological* (as opposed to purely structural or model-theoretic) aspects of semantic networks have little to do with the logic question. Semantic nets have been closely modelled on linguistic descriptions and much of the richness of natural language has been adopted. And, as I have indicated, many opportunities for efficient inference are due to the ontological structure of the world itself which is populated with objects, predicates and relations which relate to one another not with arbitrary freedom, as in unadorned logic, but rather in severely constrained ways which induce a restricted inferential hierarchy.

[82] The more venturesome might suppose that *people* experience only graphs if the ultimate structure of reality and perception is graph-theoretic; this possibility was hinted at by Kempe and Peirce and is assumed in Marty's article "Foliated Semantic Networks" in this volume.

[83] Many automatic theorem-proving methods rely on some graph-theoretic structure of scoped logical expressions (like Frege's original trees [27]) or of steps in a proof, but few use the structure of *relations* and *abstractions* characteristic of a semantic net. The sort hierarchy in an order-sorted logic *is* a semantic net.

All deductive inference involves inclusion. The truth of conclusions is included in the truth of the premises. On the other hand, all inclusion relationships can be represented by partially ordered sets (posets) and exploited for inference. Posets and lattices found in the world may be 'imported' into the inference hierarchy (as Randell & Cohn do in their article on "Exploiting"). The unifying theory for inference in Semantic Networks is thus *Order Theory* (including Lattice Theory).[84] It is by means of this theory that different posets and lattices can be combined in a principled way to produce a sound virtual inference structure which can be directly used by a machine. Many different combining operations are available along with various well-defined *completions* and derived *power-structures* in the domain of power-sets. Conversely, most complex conceptual structures can be *factored* or *reduced* into simpler constituent structures or *quotients*; these are the facets, determinables, dimensions, specialized aspects and primitive conceptual cores discussed earlier. By 1960, Lattice Theory was already recognized as important for semantic networks by a couple of the earliest AI researchers, Masterman and Parker-Rhodes (mentioned in Sections 2.2 and 12.1), but they failed to excite many others and, within AI at least, this approach has languished forgotten.[85]

Inferential posets are everywhere and it is clear that the human mind uses many of them effortlessly. The transitivity of spatial inclusion, for example, is instantly comprehended and unconsciously used, as is the poset of subsumption of meanings among fairly small relational graphs. Semantic network research should develop the principles by which all such posets may be used in the 'grand ontology' for inference. Dictionary definitions and other relations between relational graphs can be formalized with graph-grammars. I do not recommend any mathematization for its own sake, only for the power it offers. Semantic networks are analyzable with Order Theory, Graph-Grammar Theory (and some Category Theory dealing with selective preservation of relations in certain transformations), and there is promise that these theories will permit factoring most concepts in the world into compact, tractable structures built from a modest number of semantic primitives.

REFERENCES

1. **Proceedings. NOTE:** References to *Proc. IJCAI* refer to the *Proceedings* of the *International Joint Conferences on Artificial Intelligence* conducted annually by the organization of that name (IJCAI). References to *Proc. AAAI* refer to the *Proceedings of the National Conferences on Artificial Intelligence* of the American Association for Artificial Intelligence (AAAI). Both have been distributed by Morgan Kaufmann, San Mateo, California. AAAI Proceedings are now published and distributed by MIT Press, Cambridge, Mass.
2. E. Beller, Ed., *The Artificial Intelligence Compendium: Abstracts and Index to Research on AI Theory and Applications 1954-1987*, 2nd ed., Scientific Datalink, 270 Lafayette St., New York, N.Y. 10012, (1988).
3. S.C. Shapiro, Ed., *Encyclopedia of Artificial Intelligence*, John Wiley & Sons, New York, (1987).
4. R.J. Brachman, On the epistemological status of semantic networks, In *Associative Networks*, (N.V. Findler, Ed.), Academic Press, Orlando, Florida, (1979). Reprinted in *Readings in Knowledge Representation*, (R.J. Brachman and H.J. Levesque, Eds.), Morgan Kaufmann, Los Altos, California, (1985).
5. R.J. Brachman and H.J. Levesque, Eds., *Readings in Knowledge Representation*, Morgan Kaufmann, Los Altos, California, (1985).
6. D. MacRandal, Semantic networks, In *Approaches to Knowledge Representation: An Introduction*, (G.A. Ringland and D.A. Duce, Eds.), Research Studies Press/John Wiley & Sons, Letchworth, Hertfordshire/New York, (1988).
7. N. Cercone and G. McCalla, Eds., *The Knowledge Frontier: Essays in the Representation of Knowledge*, Springer, New York, (1987).
8. N. Cercone and G. McCalla, Eds., *IEEE Computer: Special Issue on Knowledge Representation*, Vol. 16, No. 10, (October 1983).
9. D.G. Bobrow and A. Collins, Eds., *Representation and Understanding: Studies in Cognitive Science*, Academic Press, Orlando, Florida, (1975).

[84] Order Theory, a generalization of Lattice Theory, addresses the formal properties of various kinds of ordered sets and their combinations, factors and generators. This is introduced in [276, 277] and documented in I. Rival's NATO series on Ordered Sets [278–280]. The mathematician's Bible of Lattice Theory is Birkhoff [281]. The mathematical journals *Order*, *Algebra Universalis* and *Discrete Mathematics* cover the subject in order of decreasing specificity.

[85] Except in recent work of Aït-Kaci and colleagues [219, 257].

10. R. Schank and K.M. Colby, Eds., *Computer Models of Thought and Language*, W.H. Freeman, San Francisco, California, (1973).
11. E. Charniak and D. McDermott, *Introduction to Artificial Intelligence*, Addison-Wesley, Reading, Mass., (1985).
12. N. Nilsson, *Principles of Artificial Intelligence*, Tioga Publishing Co., Palo Alto, California, (1980). Reprinted by Morgan Kaufmann, San Mateo, California.
13. M. Nagao, *Knowledge and Inference*, Academic Press, San Diego, California, (1990).
14. M.D. Harris, *Introduction to Natural Language Processing*, Reston Publishing Co., Reston, Va., (1985).
15. A. Barr, P. Cohen, and E. Feigenbaum, *The Handbook of Artificial Intelligence*, Vols. I–III, William Kaufmann, Palo Alto, California, (1981). Reprinted by, and Vol. IV published by, Addison-Wesley, Reading, Mass., (1989).
16. N.V. Findler, Ed., *Associative Networks: Representation and Use of Knowledge by Computers*, Academic Press, Orlando, Florida, (1979).
17. J.F. Sowa, Ed., *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, Morgan Kaufmann, San Mateo, California, (in press).
18. M.W. Evens, Ed., *Relational Models of the Lexicon: Representing Knowledge in Semantic Networks*, Cambridge University Press, Cambridge, (1988).
19. S.E. Fahlman, *NETL: A System for Representing and Using Real-World Knowledge*, MIT Press, Cambridge, Mass., (1979).
20. J.J. Sylvester, Chemistry and algebra, *Nature*, **XVII** (284,309), (1877-78). Reprinted in *The Collected Papers of James Joseph Sylvester*, Volume III, pp. 103-104, Cambridge University Press, (1909).
21. J.J. Sylvester, On an application of the new atomic theory to the graphical representation of the invariants and covariants of binary quantics—with three appendices, *Am. J. Math.*, **I**, 64–125, (1878). Reprinted in *The Collected Papers of James Joseph Sylvester*, Vol. III, pp. 148–206, Cambridge University Press, (1909).
22. A.B. Kempe, A memoir on the theory of mathematical form, *Philosophical Transactions of the Royal Society of London*, **177**, pp. 1–70, (1886). Corrected in: Note to a Memoir on the Theory of Mathematical Form, *Proceedings of the Royal Society* **42**, pp. 193–196, London, (1887).
23. A.B. Kempe, A correction and explanation, *The Monist*, **7**, 453–458, (1897).
24. D.D. Roberts, *The Existential Graphs of Charles S. Peirce*, Mouton, The Hague, (1973).
25. C.S. Peirce, *Collected Papers of Charles Sanders Peirce*, Vol. 4, (C. Hartshorne, P. Weiss and A. Burks, Eds.), Belknap Press (Harvard Univ. Press), Cambridge, Mass., (1931–1958).
26. C.S. Peirce, *The New Elements of Mathematics*, Vol. III/1, (C. Eisele, Ed.), Mouton/ Humanities Press, The Hague/Atlantic Highlands, N.J., (1976).
27. G. Frege, Begriffsschrift, eine der arthmetischen nachgebildete Formelsprache des reinen Denkens (Halle, Germany, 1879), In *From Frege to Gödel: A source book in mathematical logic, 1879–1931*, (J. van Heijenoort, Ed.), Harvard University Press, Cambridge, Mass., (1967). Title translated as: Concept-writing, a formula language, modeled upon arithmetic, for pure thought. Reprinted in *Frege and Gödel: Two Fundamental Texts in Mathematical Logic*, (J. van Heijenoort, Ed.), Harvard University Press, Cambridge, Mass., (1970).
28. K.L. Ketner, Peirce's "most lucid and interesting paper": An introduction to cenopythagoreanism, *Int. Philosophical Quarterly*, **25**, 375–392, (1986).
29. R.W. Burch, *A Peircean Reduction Thesis and the Foundations of Topological Logic*, Texas Tech Univ. Press, Lubbock, Texas, (in press).
30. H.G. Herzberger, Peirce's remarkable theorem, In *Pragmatism & Purpose: Essays Presented to Thomas A. Goudge*, (L.W. Sumner, J.G. Slater, and F. Wilson, Eds.), Univ. of Toronto Press, Toronto, (1981).
31. C.S. Hardwick, Ed., *Semiotic and Significs: The Correspondence between Charles S. Peirce and Lady Victoria Welby*, Indiana University Press, Bloomington, Indiana, (1977).
32. G. Spencer Brown, *Laws of Form*, George Allen & Unwin, London, (1969). Reprinted by Julian Press, New York, 1972, and Bantam Books (paperback), New York, 1973.
33. J.F. Sowa, *Conceptual Structures: Information Processing in Mind and Machine*, Addison-Wesley, Reading, Mass., (1984).
34. R.H. Richens, Preprogramming for mechanical translation, *Mechanical Translation*, **3** (1), (1956). (Discontinued journal. There is a complete set at the *Philip Mills Arnold Semeiology Collection*, Rare Book Department, Olin Library, Washington University, St. Louis, Mo. 63130.).
35. R.H. Richens, Report on research of the Cambridge Language Research Unit, *Mechanical Translation*, **3** (2), (1956). See previous note.
36. M. Masterman, Semantic message detection for machine translation, using an interlingua, In *1961 International Conference on Machine Translation of Languages and Applied Language Analysis—Proceedings*, Her Majesty's Stationery Office, London, (1962).
37. K. Sparck Jones, *Synonymy and Semantic Classification*, updated printing of 1964 thesis, Edinburgh University Press, Edinburgh, (1986).
38. K. Sparck Jones and B. Boguraev, A note on a study of cases, *Computational Linguistics*, **13** (1–2), 65–68, (1987).
39. S. Ceccato, *Linguistic Analysis and Programming for Mechanical Translation*, Gordon & Breach, New York, (1961).

40. W.R. Reitman, *Cognition and Thought: An Information Processing Approach*, John Wiley & Sons, New York, (1965).

41. M.R. Genesereth and N.J. Nilsson, *Logical Foundations of Artificial Intelligence*, Morgan Kaufmann, Palo Alto, California, (1988).

42. R. Turner, *Logics for Artificial Intelligence*, Ellis Horwood/John Wiley, Chichester, Sussex/New York, (1984).

43. J. Barwise and S. Feferman, Eds., *Model-Theoretic Logics*, Springer, New York, (1985).

44. W.A. Woods, What's in a link: Foundations for semantic networks, In *Representation and Understanding*, (D. Bobrow and A. Collins, Eds.), Academic Press, Orlando, Florida, (1975). Reprinted in [5].

45. R.J. Brachman, What IS-A Is and Isn't: An analysis of taxonomic links in semantic networks, *IEEE Computer*, **16**, 30–36, (1983).

46. D.S. Touretzky, *The Mathematics of Inheritance Systems*, Pitman/Morgan Kaufmann, London/San Mateo, California, (1986).

47. D. Guise, Efficient frame systems, In *Proc. of EPIA 1989*, (J.P. Martins and E.M. Morgado, Eds.), Lecture Notes in Artificial Intelligence, No. 390. Springer, Berlin, (1989).

48. Patrick J. Hayes, The logic of frames, In *Frame Conceptions and Text Understanding*, (D. Metzing, Ed.), pp. 46–61. Walter de Gruyter & Co., Berlin, (1979). Reprinted in *Readings in Knowledge Representation*, (R.J. Brachman and H.J. Levesque, Eds.), Morgan Kaufmann, Los Altos, California, (1985).

49. J.F. Allen and A.M. Frisch, What's in a semantic network? In *Proc. of the 20th Annual Meeting of the Association for Computational Linguistics*, Assoc. for Computational Linguistics, Toronto, (1982).

50. A. Deliyanni and R.A. Kowalski, Logic and semantic networks, *Communications of the ACM*, **22** (3), 184–192, (1979).

51. J.R. McSkimin and J. Minker, A predicate calculus based semantic network for deductive searching, In *Associative Networks*, (N.V. Findler, Ed.), Academic Press, Orlando, Florida, (1979).

52. M. Derthick, *An Epistomological Level Interface for Cyc*, Technical Report ACT CYC-084-90. MCC, 3500 Balcones Center Dr., Austin, Texas 78759, (1990).

53. M. Minsky, A framework for representing knowledge, In *Mind Design*, (J. Haugeland, Ed.), MIT Press, Cambridge, Mass., (1981). A longer version appeared in *The Psychology of Computer Vision*, (P.H. Winston, Ed.), McGraw-Hill, New York, (1975). Reprinted in *Readings in Knowledge Representation*, (R.J. Brachman and H.J. Levesque, Eds.), Morgan Kaufmann, Los Altos, California, (1985).

54. I.P. Goldstein and B. Roberts, Using frames in scheduling, In *Artificial Intelligence: An MIT Perspective*, (P.H. Winston, Ed.), MIT Press, Cambridge, Mass., (1979).

55. R.B. Roberts and I.P. Goldstein, *FRL Users Manual. A.I. Memo No. 408*, Artificial Intelligence Laboratory, MIT, Cambridge, Mass., (1977).

56. D.G. Bobrow and T. Winograd, An overview of KRL, a knowledge representation language, *Cognitive Science*, **1** (1), 3–46, (1977). Reprinted in *Readings in Knowledge Representation*, (R.J. Brachman and H.J. Levesque, Eds.), Morgan Kaufmann, Los Altos, California, (1985).

57. P.H. Winston and B.K.P. Horn, *LISP (1st Edition)*, Addison-Wesley, Reading, Mass., (1981).

58. P.H. Winston, *Artificial Intelligence*, Addison-Wesley, Reading, Mass., (1977).

59. Y.E. Kalay, *Modeling Objects and Environments*, John Wiley & Sons, New York, (1989).

60. R.D. Coyne, M.A. Rosenman, A.D. Radford, M. Balachandrian and J.S. Gero, *Knowledge-Based Design Systems*, Addison-Wesley, Reading, Mass., (1990).

61. R. Fikes and T. Kehler, The role of frame-based representation in reasoning, *Communications of the ACM*, **28**, 904–920, (1985).

62. B. Bruce, Case systems in natural language, *Artificial Intelligence*, **6**, 327–360, (1975).

63. C.J. Fillmore, The case for case, In *Universals in Linguistic Theory*, (E. Bach and R.T. Harms, Eds.), Holt, Reinhart & Winston, New York, (1968).

64. M. Celce-Murcia, Paradigms for sentence recognition, Technical Report, UCLA Dept. of Linguistics, published as Tech. Report afhrl-tr-72-30, Air Force Human Resources Laboratory, Training Research Division, Lowry Air Force Base, Colorado 80230, (1972).

65. R.F. Simmons, Semantic Networks: Their computation and use for understanding English sentences, In *Computer Models of Thought and Language*, (R. Schank and K.M. Colby, Eds.), W.H. Freeman, Chicago, (1973).

66. D.E. Rumelhart and D.A. Norman, Active semantic networks as a model of human memory, In *Proc. IJCAI*, (1973).

67. B.R. Ostyn, Ed., *Topics in Cognitive Linguistics*, John Benjamins, Amsterdam, (1988).

68. A. Herskovits, *Language and Spatial Cognition: An Interdisciplinary Study of the Prepositions in English*, Cambridge Univ. Press, Cambridge, (1986).

69. H.V.D. Parunak, A linguistic approach to the problem of slot semantics, In *Proc. Ann. Conf. Cognitive Science Society*, (1989).

70. R.K. Lindsay, In defense of ad hoc systems, In *Computer Models of Thought and Language*, (R. Schank and K.M. Colby, Eds.), W.H. Freeman, San Francisco, (1973).

71. Philip J. Hayes, On semantic nets, frames and associations, In *Proc. IJCAI*, (1979).

72. S.A. Safier and S. Finger, Parsing features in solid geometric models, In *Proc. ECAI-90*, 9th European Conference on Artificial Intelligence, Stockholm. Pitman, London, (1990).
73. P.H. Winston, Learning structural descriptions from examples, In *The Psychology of Computer Vision*, (P.H. Winston, Ed.), McGraw-Hill, New York, (1975). Reprinted in *Readings in Knowledge Representation*, (R.J. Brachman and H.J. Levesque, Eds.), Morgan Kaufmann, Los Altos, California, (1985).
74. H.J. Levesque and R. Brachman, A fundamental tradeoff in knowledge representation languages, In *Proc. CSCSI-84*, London, Ontario, (1984). Canadian Soc. for Computational Studies of Intelligence. Revised and reprinted in *Readings in Knowledge Representation*, (R.J. Brachman and H.J. Levesque, Eds.), Morgan Kaufmann, Los Altos, California, (1985).
75. J.G. Schmolze, Terminological knowledge representation systems supporting n-ary terms, In *Proc. KR-89, First Int. Conf. on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, California, (1989).
76. J. Adámek, *The Theory of Mathematical Structures*, D. Reidel, Dordecht, Holland, (1983).
77. P.M. Grätzer, *Universal Algebra*, Van Nostrand, Princeton, N.J., (1968).
78. D.C. Williams, The elements of being, In *Principles of Empirical Realism*, Thomas, Springfield, (1966).
79. J. Bacon, Four modelings, *J. of Philosophical Logic*, **17** (2), 91–114, (1988).
80. C. Hartshorne, A revision of Peirce's categories, *The Monist*, **63**, 277–289, (1980). Reprinted in *The Relevance of Charles Peirce*, (E. Freeman, Ed.), The *Monist* Library of Philosophy, Hegeler Institute, La Salle, Illinois, (1983).
81. P.G. Kuntz, Doing something for the categories: the choice of categorical methods and their resulting tree of categories, In *Studies in the Philosophy of Charles Sanders Peirce*, 3rd Series, (E.C. Moore and R.R. Robin, Eds.), (in press).
82. W. Percy, The Delta factor, In *The Message in the Bottle: How Queer Man Is, How Queer Language Is, and What One Has to Do with the Other*, Farrar, Straus & Giroux, New York, (1975).
83. Z. Galil, C.M. Hoffmann, E.M. Luks, C.P. Schnorr, and A. Weber, An $o(n^3 \log n)$ deterministic and an $o(n^3)$ Las Vegas isomorphism test for trivalent graphs, *J. of the ACM*, **34** (3), 513–531, (1987).
84. G. Attardi and M. Simi, Consistency and completeness of OMEGA, a logic for knowledge representation, In *Proc. IJCAI*, (1981).
85. N. Jardine and R. Sibson, *Mathematical Taxonomy*, John Wiley & Sons, London, (1971).
86. P.H.A. Sneath and R.R. Sokal, *Numerical Taxonomy*, W.H. Freeman & Co., San Francisco, (1973).
87. J. Van Ryzin, *Classification and Clustering*, Academic Press, New York, (1977).
88. R. Michalski and R. Stepp, Automatic constructions of classifications: Conceptual clustering versus numerical taxonomy, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, **5** (4), 346–409, (1983).
89. A.K. Jain and R.C. Dubes, *Algorithm for Clustering Data*, Prentice Hall, Englewood Cliffs, N.J., (1988).
90. E. Diday, Ed., *Data Analysis, Learning Symbolic and Numeric Knowledge*, Nova Science Publishers, Commack, N.Y., (1989).
91. A.A. Raskina, I.S. Sidorov, and Y.A. Shreider, Semantic foundations of object/feature languages, *Naucho-Tekhnicheskaya Informatsiya, Seriya 2* (translated as *Automatic Documentation and Mathematical Linguistics*, Allerton Press, New York), **10** (5), 18–25, (1976).
92. M.Sh. Tsalenko, Canonical representation of irreducible systems and classificational models, *Naucho-Tekhnicheskaya Informatsiya, Seriya 2* (translated as *Automatic Documentation and Mathematical Linguistics*, Allerton Press, New York), **19**, 30–34, (1985).
93. O.M. Polyakov and V.V. Dunaev, Classification schemes: Synthesis through relations, *Naucho-Tekhnicheskaya Informatsiya, Seriya 2* (translated as *Automatic Documentation and Mathematical Linguistics*, Allerton Press, New York), **19**, (5), 15–21, (1985).
94. O.M. Polyakov, A classificational data model, *Naucho-Tekhnicheskaya Informatsiya, Seriya 2* (translated as *Automatic Documentation and Mathematical Linguistics*, Allerton Press, New York), **20** (9), 13–20, (1986).
95. O.M. Polyakov, Systematization, *Naucho-Tekhnicheskaya Informatsiya, Seriya 2* (translated as *Automatic Documentation and Mathematical Linguistics*, Allerton Press, New York), **22** (12), 21–28, (1988).
96. B.P. Kononov, Problems of formal meronomy: Divisibility and incompatability of meronomic objects, *Naucho-Tekhnicheskaya Informatsiya, Seriya 2* (translated as *Automatic Documentation and Mathematical Linguistics*, Allerton Press, New York), **19** (3), 17–23, (1985).
97. J.L. Kolodner, Maintaining organization in a dynamic long-term memory, *Cognitive Science*, **7**, 243–280, (1983).
98. J.L. Kolodner, *Retrieval and Organizational Strategies in Conceptual Memory: A Computer Model*, Lawrence Erlbaum Associates, Hillside, N.J., (1984).
99. P. Koton and M.P. Chase, Knowledge representation in a case-based reasoning system: Defaults and exceptions, In *Proc. KR-89, 1st Int. Conf. on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, California, (1989).
100. D.J. Israel, Interpreting network formalisms, *Computers & Math. w. Applications*, **9**, 1–13, (1983). Reprinted in *Computational Linguistics*, (N. Cercone, Ed.), Pergamon, Oxford, (1983).
101. P.F. Patel-Schneider, What's inheritance got to do with knowledge representation?, In Maurizio Lenzerini, Daniele Nardi, and Maria Simi, Editors, *Inheritance Hierarchies in Knowledge Representation and Programming Languages*, John Wiley & Sons, New York, (1991).

102. A.M. Frisch and A.G. Cohn, Thoughts and afterthoughts on the 1988 Workshop on Principles of Hybrid Reasoning, *AI Magazine* (Special Workshop Issue), **11** (5), 77–87, (1991).

103. O. Senyk, R.S. Patil, and F.A. Sonnenberg, Systematic knowledge base design for medical diagnosis, *J. of Applied Artificial Intelligence*, **3**, 249–274, (1989). Reprinted in *Causal Artificial Intelligence Models: Steps toward Application*, (W. Horn, Ed.), pp. 165–189, Hemisphere Publishing, New York, (1989).

104. R.V. Guha and D.B. Lenat, Cyc: a midterm report, *AI Magazine*, **11** (3), 32–59, (1990).

105. M.L. Ginsberg, Ed., *Readings in Nonmonotonic Reasoning*, Morgan Kaufmann, San Mateo, California, (1897).

106. B. Nebel, Reasoning and revision in hybrid representation systems, Ph.D. Thesis, Univ. of Saarland, Saarbrücken, Germany, (1989). Reprinted, Springer Lecture Notes on Artificial Intelligence No. 422, Springer, Berlin, (1989).

107. J. Pearl, *Probabilistic Reasoning in Intelligent Systems*, Morgan Kaufmann, San Mateo, California, (1988).

108. F. Bacchus, A modest, but semantically well founded, inheritance reasoner, In *Proc. IJCAI*, (1989).

109. M. Vardi, Ed., *Proc. 2nd Conf. on Theoretical Aspects of Reasoning about Knowledge*, Morgan Kaufmann, San Mateo, California, (1988).

110. R.P. Loui, M. Kahn, and G. Simari, Workshop on defeasible reasoning with specificity and multiple inheritance, Technical Report WUCS-90-22, Dept. of Computer Science, Washington University, St. Louis, MO 63130, (1990).

111. J. Dorosh and R.P. Loui, Edited transcription of the workshop on defeasible reasoning with specificity and multiple inheritance, *SIGART Bulletin*, **2** (1), 3–51, (1991).

112. R.V. Guha, The representation of defaults in Cyc, In *Proc. AAAI-90*, (1990).

113. R. Chaffin and D.J. Herrmann, The nature of semantic relations: A comparison of two approaches, In *Relational Models of the Lexicon*, (M.W. Evens, Ed.), Cambridge Univ. Press, Cambridge, (1988).

114. R. Chaffin and D.J. Herrmann, Relation element theory: A new account of the representation and processing of semantic relations, In *Memory and Learning: The Ebbinghaus Centennial Conference*, (D. Gorfein and R. Hoffman, Eds.), Lawrence Erlbaum Associates, Hillsdale, N.J., (1987).

115. P.R. Cohen and C. Loiselle, Beyond IS-A: Structures for plausible inference in semantic networks, In *Proc. AAAI-88*, (1988).

116. M. Vilain, The restricted language architecture of a hybrid representation system, In *Proc. IJCAI*, (1985).

117. M.G. Moser, An overview of NIKL, the new implementation of KL-ONE, In Research in knowledge representation for natural language understanding—Annual report 1982–1983, Technical Report 5421, BBN Laboratories, (1983).

118. M. Derthick, Counterfactual reasoning with direct models, In *Proc. AAAI-87*, (1987).

119. R.H. Thomason and D.S. Touretzky, Inheritance theory and networks with roles, In *Principles of Semantic Networks*, (J.F. Sowa, Ed.), Morgan Kaufmann, San Mateo, California, (in press).

120. M.N. Huhns and L.M. Stephens, Plausible inferencing using extended composition, In *Proc. IJCAI*, (1989).

121. M.N. Huhns and L.M. Stephens, Extended composition of relations, MCC Technical Report ACA-AI-376-88, MCC, 3500 W. Balcones Center Dr., Austin, Texas 78759, (1988).

122. R.R. Bakker, Knowledge graphs: Representation and structuring of scientific knowledge, Ph.D. Thesis, University of Twente, Twente, Netherlands, (1987).

123. J.R. Burns, W.H. Winstead, and D.A. Haworth, Semantic nets as paradigms for both causal and judgemental knowledge representation, *IEEE Trans. on Systems, Man and Cybernetics*, **19** (1), 58–67, (1989).

124. R. Davis and D.B. Lenat, *Knowledge-Based Systems in Artificial Intelligence*, McGraw-Hill, New York, (1982).

125. R. Greiner and D.B. Lenat, A representation language language, In *Proc. AAAI-80*, (1980).

126. D.B. Lenat and R.V. Guha, *Building Large Knowledge Based Systems*, Addison-Wesley, Reading, Mass., (1990).

127. G.G. Hendrix, Encoding knowledge in partitioned networks, In *Associative Networks*, (N.V. Findler, Ed.), Academic Press, Orlando, Florida, (1979).

128. D. Harel, On visual formalisms, *Communications of the ACM*, **31**, 514–530, (1988).

129. J.W. Esch, Graphical displays for conceptual structures, In *Proc. 5th Annual Workshop on Conceptual Structures*, (L. Gerholz and P. Elkund, Eds.), J.W. Esch, Unisys, Box 64525, St. Paul, Minnesota 55164, (1990).

130. W.A. Woods, Understanding subsumption and taxonomy: A framework for progress, In *Principles of Semantic Networks*, (J.F. Sowa, Ed.), Morgan Kaufmann, San Mateo, California, (in press).

131. H.J. Levesque, Making believers out of computers, *Artificial Intelligence*, **30**, 81–108, (1986). Reprinted in *Readings in Artificial Intelligence and Database Systems*, (J. Mylopoulos and M.L. Brodie, Eds.), Morgan Kaufmann, San Mateo, California, (1989).

132. D.W. Etherington, A. Borgida, R.J. Brachman and H. Kautz, Vivid knowledge and tractable reasoning: Preliminary report, In *Proc. IJCAI*, (1989).

133. M.G. Dyer, Symbolic neuroengineering for natural language processing: A multilevel research aproach, In *Advances in Connectionist and Neural Computation Theory, Volume 1: High-Level Connectionist Models*, (J.A. Barnden and J.B. Pollock, Eds.), Ablex, Norwood, N.J., (1991).

134. J.L. Borges, The Analytical Language of John Wilkins, In *Other Inquisitions: Essays 1937-1952*, Univ. of Texas Press, Austin, Texas, (1964).

135. B. Smith and K. Mulligan, Pieces of a theory, In *Parts and Moments: Studies in Logic and Formal Ontology*, (B. Smith, Ed.), Philosophia, Munich, (1986). The latter part, beginning with Section 6.4, p. 81, based on work by Wolfgang Degen at the Univ. of Erlangen. See also "Framework for Formal Ontology" in *Topoi* 3, (1982).

136. J. McCarthy, *Situations, Actions, and Causal Laws*, Stanford AI Project Memo No. 2, Stanford University, Stanford, California, (1963).

137. J. McCarthy and P. Hayes, Some philosophical problems from the standpoint of artificial intelligence, *Machine Intelligence*, 4, 463–502, (1969). Reprinted in *Readings in Nonmonotonic Reasoning*, (M.L. Ginsberg, Ed.), Morgan Kaufmann, San Mateo, California, (1987).

138. E. Davis, *Representations of Commonsense Knowledge*, Morgan Kaufmann, San Mateo, California, (1990).

139. J. Barwise and J. Perry, *Situations and Attitudes*, MIT Press, Cambridge, Mass., (1983).

140. W.M. Tepfenhart and A.V. Lazzara, The situation data model, In *Proc. 3rd Ann. Conf. on Conceptual Graphs*, (J.W. Esch, Ed.). This and later papers are available from the authors, Knowledge Systems Concepts, 262 Liberty Plaza, Rome, N.Y. 13440-5812, (1988).

141. R. Montague, *Formal Philosophy: Selected Papers of Richard Montague*, (R. Thomason, Ed.), Yale Univ. Press, New Haven, Conn., (1974).

142. R.M. Martin, *Semiotics and Linguistic Structure*, State Univ. of New York Press, Albany, New York, (1978).

143. L.K. Schubert and C.H. Hwang, An episodic knowledge representation for narrative texts, In *Proc. KR-89, First Int. Conf. on Principles of Knowledge Representation and Reasoning, Toronto*, Morgan Kaufmann, San Mateo, California, (1989).

144. L.K. Schubert and C.H. Hwang, An episodic knowledge representation for narrative texts, (longer version, 53 pp.) Technical Report 345, TR Librarian, Dept. of Computer Science, Univ. of Rochester, Rochester, N.Y. 14627, (1990).

145. W. Lehnert, Knowledge based natural language understanding, In *Exploring Artificial Intelligence*, (H.E. Shrobe and AAAI, Eds.), Morgan Kaufmann, San Mateo, California, (1988).

146. D. Soergel, *Indexing Languages and Thesauri: Construction and Maintainance*, Melville Publishing (John Wiley & Sons), Los Angeles, (1974).

147. D.B. Lenat, R.V. Guha, K. Pittman, D. Pratt and M. Shepherd, Cyc: Toward programs with common sense, *Communications of the ACM*, 33 (8), 30–49, (1990).

148. J.F. Sowa, Knowledge acquisition by teachable systems, In *Proc. EPIA 80—4th Portugese Conference on Artificial Intelligence*, (J.P. Martins and E.M. Morgado, Eds.), *Lecture Notes in Artificial Intelligence* No. 390, Springer, Berlin, (1990).

149. J.L. Jolley, The Holotheme, In *Conceptual Basis of the Classification of Knowledge: Proc. of the Ottawa Conf.*, (J.A. Wojciechowski, Ed.), Verlag Dokumentation, Box 148, D-8023 Pullach bei München, Germany, (1971).

150. H.G. Burger, *The Wordtree*, Wordtree Publ., 10876 Bradshaw, Overland Park, Kansas 66210-1148, (1984). Privately printed. There is a copy at the *Philip Mills Arnold Semeiology Collection*, Rare Book Department, Olin Library, Washinton University, St. Louis, MO. 63130.

151. P. Doudna, Personal communication, (1990).

152. C. Rieger and M. Grinberg, The declarative representation and procedural simulation of of causality in physical systems, In *Proc. IJCAI*, (1977).

153. D.G. Bobrow, Ed., *Qualitative Reasoning about Physical Systems*, Special Issue of *Artificial Intelligence* journal, 24, (1984). Reprinted MIT Press, Cambridge, Mass., (1985).

154. D.S. Weld and J. de Kleer, Eds., *Readings in Qualitative Reasoning about Physical Systems*, Morgan Kaufmann, San Mateo, California, (1989).

155. R.S. Patil, Techniques for diagnostic reasoning in medicine, In *Exploring Artificial Intelligence*, (H.E. Shrobe and AAAI, Eds.), Morgan Kaufmann, San Mateo, California, (1988).

156. R.S. Patil, Causal representation of patient illness for electrolyte and acid-base diagnosis, Technical Report No. 267, MIT Laboratory for Computer Science, 545 Technology Square, Cambridge, Mass. 02189, (1981).

157. Y. Shoham, *Reasoning about Change: Time and Causation from the Standpoint of Artificial Intelligence*, MIT Press, Cambridge, Mass., (1988).

158. W. Zadrożny, Networks of signs for prospectors of minds, (unpublished MS.)

159. R.C. Gonzales and M.G. Thomason, *Syntactic Pattern Recognition: An Introduction*, Addison-Wesley, Reading, Mass., (1978).

160. K.S. Fu, *Syntactic Methods in Pattern Recognition*, Academic Press, New York, (1974).

161. T.L. Kunii, Ed., *Visual Database Systems*, North-Holland, Amsterdam, (1989).

162. L. Bolc, Ed., *Natural Language Communication with Pictorial Information Systems*, Springer, Berlin, (1984).

163. B. Neumann, Natural language description of time-varying scenes, In *Semantic Structures: Advances in Natural Language Processing*, (D. Waltz, Ed.), Lawrence Erlbaum Associates, Hillsdale, New Jersey, (1989).

164. W.C. Bulko, Understanding coreference in a system for solving physics word problems, Ph.D. Thesis, University of Texas, Austin, Texas 78712, (1989).

165. F. Giunchiglia and E. Trucco, Object configuration by incremental ill-described spatial constraints, Technical Report No. 400, Dept. of Artificial Intelligence, University of Edinburgh, 80 S. Bridge, Edinburgh EH1 1HN, Scotland, (1988).

166. A.R. Rao and R. Jain, Knowledge representation and control in computer vision systems, *IEEE Expert*, 64–79, (Spring 1988).

167. D.H. Ballard, An approach to knowledge-directed image analysis, In *Computer Vision Systems*, (A.R. Hanson and E.M. Riseman, Eds.), Academic Press, New York, (1978).

168. A. Guzman, Analysis of curved line drawings using context and global information, In *Machine Intelligence 6*, (B. Meltzer and D. Michie, Eds.), Edinburgh University Press, Edinburgh, (1971).

169. Y. Lirov, E.Y. Rodin, B.G. McElhany and L.W. Wilbur, Artificial intelligence modelling of control systems, *Simulation*, **50** (1), 12–24, (1988).

170. E.Y. Rodin, Semantic control theory, *Applied Mathematics Letters*, **1**(1), 73–78, (1988).

171. J. Sztipanovits, A computation model for semantic control, Technical Report, May 89, Dept. of Elec. Eng., Vanderbilt Univ., Nashville, Tenn. 37235, (1989).

172. M.D. Mesarović, On self-organizational systems, In *Self-Organizing Systems 1962*, (M.C. Yovits, G.T. Jacobi, and G.P. Goldstein, Eds.), Spartan Books, Washington, D.C., (1962).

173. L. Acar and Ü. Özgüner, Design of knowledge-rich hierarchical controllers for large functional systems, *IEEE Trans. on Systems, Man & Cybernetics*, **20** (4), 791–803, (1990).

174. J.S. Hollan, E.L. Hutchins, and L. Weitzman, Steamer: An interactive inspectable simulation-based training system, *AI Magazine*, **5** (2), 15–27, (1984).

175. H.W. Beck and P.A. Fishwick, Incorporating natural language descriptions into modelling and simulation, *Simulation*, **52** (3), 95–101, (1989).

176. P.A. Fishwick, HIRES: Hierarchical reasoning system version 1.0—User's manual, Technical Report 1989, Dept. of Comp. Science, Univ. of Pennsylvania, Philadelphia, Penn., (1985).

177. L. Joskowicz, Mechanism comparison and classification for redesign, Research Report No. RC 14573(# 65152) 4/19/89, IBM T.J. Watson Research Center, Yorktown Heights, N.Y. 10598, (1989).

178. N.V. Findler, A heuristic information retrieval system based on associative networks, In *Associative Networks*, (N.V. Findler, Ed.), Academic Press, Orlando, Florida, (1979).

179. N.V. Findler, *An Artificial Intelligence Technique for Information and Fact Retrieval—An Application in Medical Knowledge Processing*, MIT Press, Cambridge, Mass., (to appear).

180. J.M. Crawford and B. Kuipers, ALL: formalizing access limited reasoning, In *Principles of Semantic Networks*, (J.F. Sowa, Ed.), Morgan Kaufmann, San Mateo, California, (in press).

181. J.M. Crawford and B. Kuipers, Towards a formalization of access limited logic, Technical Report AI90-133, Univ. of Texas, Austin, Texas 78712, (1990).

182. P.H. Winston, Learning and reasoning by analogy, *Communications of the ACM*, **23** (12), (Dec. 1980).

183. D. Gentner, Structure-mapping: A theoretical framework for analogy, *Cognitive Science*, **7**, 155–170, (1983).

184. D. Leishman, Analogy as a constrained partial correspondence over conceptual graphs, In *Proc. KR-89, 1st Int. Conf. on Principles of Knowledge Representation and Reasoning*. Morgan Kaufmann, San Mateo, California, (1989).

185. E.C. Way, Dynamic type hierarchies: An approach to knowledge representation through metaphor, Ph.D. Thesis, State Univ. of New York at Binghamton, (1987). Published as *Knowledge Representation and Metaphor*, Kluwer, Dordecht, Holland, (1991).

186. B.K. Boguraev and E.J. Briscoe, *Computational Lexicography for Natural Language Processing*, Longman, London, (1989).

187. M.R. Quillian, Word concepts, *Behavioral Science* **12**, 410–430, (1967). Reprinted in *Readings in Knowledge Representation*, (R.J. Brachman and H.J. Levesque, Eds.), Morgan Kaufmann, Los Altos, California, (1985).

188. M.R. Quillian, Semantic memory, In *Semantic Information Processing*, (M. Minsky, Ed.), MIT Press, Cambridge, Mass., (1968).

189. I. Georgescu, The hypernets method for representing knowledge, In *Artificial Intelligence: Methodology, Systems, Applications*, (W. Bibel and B. Petkoff, Eds.), North-Holland, Amsterdam, (1985).

190. Y. Wilks, Good and bad arguments about semantic primitives, *Communications and Cognition* **10**, 181–221, (1977).

191. T. Winograd, On primitives, prototypes, and other semantic anomalies, In *Proc. TINLAP-2*, (D.L. Waltz, Ed.), 2nd Conf. on Theoretical Issues in Natural Language Processing. Coordinated Science Laboratory, Univ. of Illinois, Urbana, Illinois, (1978).

192. R. Llull, *Ars Brevis*. Barcelona, (1308). Translated in *Selected Works of Ramon LLull*, Vol. II, (A. Bonner, Ed.), Princeton Univ. Press, Princeton, N.J., (1985).

193. M. Gardner, *Logic Machines and Diagrams*, 2nd ed., Univ. of Chicago Press, Chicago, Illinois, (1982).

194. G.W. Leibniz, A calculus of consequences, In *Philosophical Essays*, (R. Ariew and D. Garber, Eds.), Hackett Publ. Co., Indianapolis, Indiana, (1989).

195. K. Gödel, Über Formal Unentscheidbare Sätze der *Principia Mathematica* und Verwandter Systeme I, *Monatshefte für Mathematik und Physik*, **38**, 173–198, (1931). Translated as: 'On Formally Undecidable Propositions of *Principia Mathematica* and Related Systems' in *From Frege to Gödel: A source book in mathematical*

logic, 1879-1931, (J. van Heijenoort, Ed.), Harvard Univ. Press, Cambridge, Mass., (1967), and reprinted in *Frege and Gödel: Two Fundamental Texts in Mathematical Logic*, (J. van Heijenoort, Ed.), Harvard Univ. Press, Cambridge, Mass., (1970).

196. J. Wilkins, *An Essay Towards a Real Character and a Philosophical Language*, The Royal Society, printed for Gellibrand by Martin, London, (1668). Reprinted by Scolar Press, Menston, Yorkshire, England, (1968).
197. M.M. Slaughter, *Universal Languages and Taxonomy in the Seventeenth Century*, Cambridge Univ. Press, Cambridge, (1982).
198. P. Cornelius, *Languages in Seventeenth and Early Eighteenth Century Imaginary Voyages*, Librarie Droz, Geneva, (1965). Describes languages of Dalgarno, Wilkins and Leibniz, and purported central Australian languages, all using semantic primitives.
199. J. Cohen, On the project of a universal character, *Mind*, New Series **63** (49), 49–63, (1954).
200. F. Lodwick, *A Common Writing: Whereby, Two Although not Understanding the Others Language, Yet by the Helpe thereof, May Communicate Their Minds to One Another*, London, (1647), In *The Works of Francis Lodwick*, (V. Salmon, Ed.), Longman, London, (1972).
201. G. Dalgarno, *Ars Signorum, vulgo Character Universalis et Lingua Philosophica*, J. Hayes, London, (1661). Reprinted in *The Works of George Dalgarno of Aberdeen*, the Maitland Club, Edinburgh, (1834); reprinted by AMS Press/Johnson Reprint Corp., New York, (1971).
202. Anonymous, *The Description and Explanation of a "Universal Character" or, Manner of Writing, that may be Intelligble to the Inhabitants of Every Country, although Ignorant of Each Others Language; and which is to be Learnt with Facility because Founded on a Simple and Easy Mode of Classifying Our Ideas, and Requiring But Few Arbitrary Signs*, J. Hollway, Bath, England, (*circa* 1835).
203. Anonymous, *Die Neue Sinnschrift 'Safo' als Einheitszeichenschrift der Völker*, Private Edition, Stamberg, Germany, (1956).
204. C.K. Bliss, *International Semantography, a Non Alphabetical Symbol Writing Readable in All Languages*, Vols. 1, 2 & 3. Privately Printed, Sydney, (1949). Revised edition (1965). Material distributed by Blissymbolics Institute, 350 Rumsey Rd., Toronto, Ontario M4G 1R8, Canada, and EBSCO, 1230 1st Ave. N., Birmingham, Alabama 35203.
205. E. Helfman, *Blissymbolics—Speaking without Speech*, Elsevier, New York, (1981).
206. A.N. Whitehead and B. Russell, *Principia Mathematica*, 2nd ed., Cambridge Univ. Press, Cambridge, (1927).
207. Patrick J. Hayes, The second naive physics manifesto, In *Formal Theories of the Commonsense World*, (J.R. Hobbs and R.C. Moore, Eds.), Ablex, Norwood, N.J., (1985). Reprinted in *Readings in Knowledge Representation*, (R.J. Brachman and H.J. Levesque, Eds.), Morgan Kaufmann, Los Altos, California, (1985).
208. H.A. Freudenthal, *LINCOS: Design of a Language for Cosmic Intercourse*, North-Holland, Amsterdam, (1960).
209. A. Wierzbicka, *Lingua Mentalis*, Academic Press, Sydney/New York, (1980).
210. A. Wierzbicka, *English Speech Act Verbs: A Semantic Dictionary*, Academic Press, Sydney/New York, (1989).
211. J. Laffal, *A Concept Dictionary of English*, Gallery Press, Essex, Conn./John Wiley & Sons, New York, (1973).
212. I. Mel'čuk and A. Zholkovsky, *The Explanatory Combinatorial Dictionary of Modern Russian. Semantico-Syntactic Studies of Russian Vocabulary*, Wiener Slawistischer Almanach, Vienna, (1984). Relevant Introduction translated in *Relational Models of the Lexicon*, (M.W. Evens, Ed.), Cambridge Univ. Press, Cambridge, (1988).
213. K.C. Litkowski, On the search for semantic primitives, *Computational Linguistics*, **14** (1), 52, (1988).
214. K.C. Litkowski, Models of the semantic structure of dictionaries, *American Journal of Computational Linguistics*, Microfiche **81**, 25–74, (1978).
215. K.C. Litkowski, Models of the semantic structure of dictionaries, In *Proc. 18th Ann. Meeting of the Assoc. for Computational Linguistics*, (1978). Extended abstract. Orig. Unpubl. MS.: K. Litkowski, 20239 Lea Pond Pl., Gaithersburg, MD. 20879.
216. A.F. Parker-Rhodes, *Inferential Semantics*, Harvester Press/Humanities Press, Hassox, Sussex/Atlantic Highlands, N.J., (1978).
217. A.F. Parker-Rhodes, A new model of syntactic description, In *1961 International Conference on Machine Translation of Languages and Applied Language Analysis—Proceedings*, London, (1962). Her Majesty's Stationery Office.
218. P. Doudna, Solid state logic, (Three-dimensional cardboard rhombic dodecahedral lattice of quantifiers.) P. Doudna, 10644 Jesskamp Dr., St. Louis, MO. 63136, (1990).
219. H. Aït-Kaci, *A Lattice Theoretic Approach to Computation Based on a Calculus of Partially Ordered Type Structures*, Ph.D., Thesis, University of Pennsylvania, (1984).
220. M. Kay, Functional grammar, In *Proc. 5th Annual Meeting of the Berkeley Linguistics Society*, Berkeley, California, (1979). Berkeley Linguistics Soc.
221. M. Kay, Parsing in functional grammar, In *Natural Language Parsing*, (D. Dowty, L. Karttunen and A. Zwicky, Eds.), Cambridge Univ. Press, Cambridge, (1985).
222. K. Knight, Unification: A multidisciplinary survey, *ACM Computing Surveys*, **21** (1), 93–124, (1989).

223. H. Aït-Kaci and R. Nasr, LOGIN: A logic programming language with built-in inheritance, *J. of Logic Programming*, **3**, 185–215, (1986).

224. G. Smolka and H. Aït-Kaci, Inheritance hierarchies: Semantics and unification, In *Unification*, (C. Kirchner, Ed.), Academic Press, London, (1990).

225. K.H. Bläsius, U. Hedtstück, and C.-R. Rollinger, Eds., *Sorts and Types in Artificial Intelligence*, Lecture Notes In Artificial Intelligence, No. 418. Springer, Berlin, (1990).

226. G. Smolka, *A Feature Logic with Subsorts*, LILOG Report 33, IBM Deutschland, Postfach 80 08 80, 7000 Stuttgart 80, Germany, (1988).

227. B. Nebel and G. Smolka, Representation and reasoning with attributive descriptions, In *Sorts and Types in Artificial Intelligence*, (K.H. Bläsius, U. Hedtstück, and C.-R. Rollinger, Eds.), Lecture Notes In Artificial Intelligence, No. 418. Springer, Berlin, (1990).

228. M. Schmidt-Schauß, Subsumption in KL-ONE is undecidable, In *Proc. KR-89, 1st Int. Conf. on Principles of Knowledge Representation and Reasoning*, Morgan Kaufmann, San Mateo, California, (1989).

229. W. Kent, Limitations of record-based information models, *ACM Transactions on Database Systems*, **4**, 108–131, (1979). Reprinted in *Readings in Artificial Intelligence and Databases*, (Mylopoulos and Brodie, Eds.), Morgan Kaufmann, San Mateo, California, (1989).

230. J. Searle, Determinables and the notion of resemblance, *Proc. of the Aristotelian Society*, Supplementary Vol. **33**, 141–158, (1959).

231. W.E. Johnson, *Logic*, Dover, New York, (1964).

232. A. Hautamäki, *Points of View and Their Logical Analysis*, Acta Philosophica Fennica No. 41. Societas Philosophica Fennica, Academic Bookstore, Keskuskatu 1, 00100 Helsinki 10, Finland, (1986).

233. W.S. Torgerson, *Theory and Methods of Scaling*, John Wiley & Sons, New York, (1958).

234. B. Ganter and R. Wille, Conceptual scaling, In *Applications of Combinatorics and Graph Theory to the Biological and Social Sciences*, (F. Roberts, Ed.), Springer, New York, (1989).

235. V.Sh. Rubashkin, Attribute and value, *Naucho-Tekhnicheskaya Informatsiya, Seriya 2* (translated as *Automatic Documentation and Mathematical Linguistics* published by Allerton Press, New York), **3**, 3–10, (1976).

236. K. Lehmann, *Dimensional Abstraction and Structure in Tonal Music*, Master's Thesis, Univ. of Washington, Seattle, Washington, (1989).

237. J.A. Wojciechowski, Ed., *Conceptual Basis of the Classification of Knowledge: Proc. of the Ottawa Conf.*, Verlag Dokumentation, Box 148, D-8023 Pullach bei München, Germany, (1971).

238. C.D. Batty, *An Introduction to Colon Classification*, Archon Books, Hamden, Conn., (1967).

239. S.R. Ranganathan, *The Colon Classification*, 5th Ed., Madras Library Assn., Madras, India, (1957), 6th Ed., Graduate School of Library Science, Rutgers State Univ., New Brunswick, N.J., (1965).

240. G. Bhattacharyya and S.R. Ranganathan, From knowledge classification to library classification, In *Conceptual Basis of the Classification of Knowledge: Proc. of the Ottawa Conf.*, (J.A. Wojciechowski, Ed.), Verlag Dokumentation, Box 148, D-8023 Pullach bei München, Germany, (1971).

241. S.R. Ranganathan, *Prolegomena to Library Classification*, 3rd Ed., Madras, India, (1967).

242. R. Prieto-Diaz, Implementing faceted classification for software reuse, *Communications of the ACM*, **34** (5), 89–97, (1991).

243. O. Mariño, F. Rechenmann, and P. Uvietta, Multiple perspectives and classification mechanism in object-oriented representation, In *Proc. ECAI-90*, 9th European Conf. on Artificial Intelligence, Stockholm, Pitman, London, (1990).

244. M.J. Bates, Subject access in online catalogs: A design model, *J. of the American Soc. of Information Science*, **37**(6), 357–375, (1986).

245. G. Ellis, Compiling conceptual graphs, In *6th Annual Workshop On Conceptual Graphs*, SUNY Binghamton, Binghamton, N.Y., (to appear, 1991).

246. P.P.-S. Chen, The entity-relationship model—toward a unified view of data, *ACM Trans. on Database Systems*, **1** (1), 9–36, (1976). Reprinted in *Readings in Artificial Intelligence and Databases*, (J. Mylopoulos and M. Brodie, Eds.), Morgan Kaufmann, Palo Alto, California, (1989).

247. G. Nijssen, F. van Assche and J. Snijders, End user tools for information systems requirement definition, In *Formal Models and Practical Tools for Information System Design*, (H. Schneider, Ed.), North-Holland, Amsterdam, (1979).

248. G. Verheijen and J.V. Bekkum, NIAM: An Information Analysis Method, In *Information System Design Methodologies: A Comparative Review*, (T. Olle, H. Sol, and A. Verrijn-Stuart, Eds.), IFIP, North-Holland, Amsterdam, (1982).

249. E. Sandewall, Theory of information management systems, Technical Report No. LiTH-IDA-R-83-03, Dept. of Computer and Information Science, Linköping Univ., S-581 83 Linköping, Sweden, (1983).

250. R. Rönnquist, Network and lattice based approaches to the representation of knowledge, Technical Thesis No. 113, Dept. of Computer & Information Science, Linköping University, S-581 83 Linköping, Sweden, (1987).

251. D.C. Tsichritzis and F.H. Lochovsky, *Data Models*, Prentice-Hall, Englewood Cliffs, N.J., (1982).

252. J. Mylopoulos and M.L. Brodie, Eds., *Readings in Artificial Intelligence and Database Systems*, Morgan Kaufmann, San Mateo, California, (1989).

253. M.L. Brodie, J. Mylopoulos, and J.W. Schmidt, Eds., *On Conceptual Modelling: Perspectives from Artificial Intelligence, Databases and Programming Languages*, Springer, New York, (1984).

254. M.L. Brodie and J. Mylopoulos, Eds., *On Knowledge Based Management Systems*, Springer, New York, (1986).

255. S.B. Zdonik and D. Maier, Eds, *Readings in Object-Oriented Database Systems*, Morgan Kaufmann, San Mateo, California, (1990).

256. W. Kim and F.H. Lochovsky, Eds., *Object-Oriented Concepts, Databases, and Applications*, ACM Press/Addison-Wesley, New York/Reading, Mass., (1989).

257. H. Aït-Kaci, R. Boyer, P. Lincoln, and R. Nasr, Efficient implementation of lattice operations, *ACM Trans. on Programming Languages and Systems* **11**, 115–146, (1989).

258. G. Hinton and J.A. Anderson, *Parallel Models of Associative Memory*, Lawrence Erlbaum Associates, Hillsdale, N.J., (1989). Update of 1981 edition.

259. J.A. Barnden and J.B. Pollack, Eds., *Advances in Connectionist and Neural Computation Theory, Volume 1: High-Level Connectionist Models*, Ablex, Norwood, N.J., (1991).

260. G.E. Hinton, Representing part-whole hierarchies in connectionist networks, In *Proc. of the Cognitive Science Society*, (1988).

261. D.S. Touretzky and G.E. Hinton, A distributed connectionist production system, In *Proc. of the 9th Annual Conf. of the Cognitive Science Society*. Also in *Cognitive Science* **12** (3), 423-466, (1987).

262. G. Pinkas, Energy minimization and the satifiability of propositional calculus, In *Proc. of the 1990 Connectionist Models Summer School*, (D.S. Touretzky, J.L. Elman, T.J. Sejnowski, and G.E. Hinton, Eds.), Morgan Kaufmann, San Mateo, California, (1990).

263. G. Pinkas, Propositional non-monotonic reasoning and inconsistency in symmetric neural networks, In *Proc. IJCAI*, (1991).

264. T. van Gelder, Compositionality: A connectionist variation on a classical theme, *Cognitive Science*, **14** (3), 355–384, (1990).

265. G.E. Hinton, Implementing semantic networks in parallel hardware, In *Parallel Models of Associative Memory*, (G.E. Hinton and J.A. Anderson, Eds.), Lawrence Erlbaum Associates, Hillsdale, N.J., (1989). Update of 1981 version.

266. J. Conklin, Hypertext: An introduction and survey, *IEEE Computer*, **20** (9), 17, (1987).

267. T. Richards, *Clausal Form Logic*, Addison-Wesley, Sydney, (1989).

268. P. Wegner, Dimensions of object-based language design, In *Proc. of the Conf. on Object-Oriented Systems, Languages and Applications (OOPSLA)*, (1987).

269. J.D. McGregor and T. Korson, Special Issue on Object-Oriented Design, *Communications of the ACM*, **33** (9), (1990).

270. B. Shriver and P. Wegner, Eds., *Research Directions in Object-Oriented Programming*, MIT Press, Cambridge, Mass., (1987).

271. M. Lenzerini, D. Nardi, and M. Simi, Eds., *Inheritance Hierarchies in Knowledge Representation and Programming Languages*, John Wiley & Sons, New York, (1991).

272. B.J. Fladung, *The XLISP Primer*, Prentice-Hall, Englewood Cliffs, N.J., (1987). XLISP available from AI SIG, Boston Computer Society, 1 Center Plaza, Boston, Mass. 02108.

273. D.J. Israel and R.J. Brachman, Some remarks on the semantics of representation languages, In *On Conceptual Modelling*, (M.L. Brodie, J. Mylopoulos, and J.W. Schmidt, Eds.), Springer, New York, (1984).

274. J.F. Sowa, Matching logical structure to linguistic structure, In *Studies in the Logic of Charles Sanders Peirce*, (N. Houser, D.D. Roberts, and J. Van Evra, Eds.), Indiana Univ. Press, Bloomington, Indiana, (to appear) (1991).

275. L.K. Schubert, Semantic networks are in the eye of the beholder, In *Principles of Semantic Networks*, (J.F. Sowa, Ed.), Morgan Kaufmann, San Mateo, California, (in press).

276. B.A. Davey and H.A. Priestley, *Introduction to Lattices and Order*, Cambridge Univ. Press, Cambridge, (1990).

277. H. Gericke, *Lattice Theory*, George G. Harrap/Frederick Ungar, London/New York, (1966). Originally published as *Theorie der Verbände*, Mannheim, Germany, (1963).

278. I. Rival, Ed., *Ordered Sets*, NATO ASI Series No. 83, Reidel, Dordecht, Netherlands, (1982).

279. I. Rival, Ed., *Graphs and Order*, NATO ASI Series No. 147, Reidel, Dordecht, Netherlands, (1985).

280. I. Rival, Ed., *Algorithms and Order*, NATO ASI Series No. 255, Reidel, Dordecht, Netherlands, (1989).

281. G. Birkhoff, *Lattice Theory*, 3rd Ed., Coll. Publ. XXV, American Math. Soc., Providence, R.I., (1967).