
Much combinatory reasoning can be short-circuited by using special graphical and geometrical methods when determining certain relationships—the ones people grasp without thinking.

Determining Type, Part, Color, and Time Relationships

Lenhart K. Schubert, Mary Angela Papalaskaris, and Jay Taugher

University of Alberta, Edmonton

At the University of Alberta, we are trying to construct a system with enough commonsense knowledge and fluency in English to be able to answer simple questions about a wide range of mundane subjects.* Since we would like the system to remain comprehensible and efficient no matter how large or varied its knowledge base, our emphasis has been on theoretically well-founded design at all stages.^{1,2}

Several kinds of inference problems arise constantly in question-answering processes, which, without special handling, can absorb large computational resources. One kind requires determining how two types of things are related, e.g., whether person subsumes girl, or whether girl is incompatible with computer; others require determining these or similar relationships among parts of objects, colors, or times.

This collection of special inference domains is not as haphazard as it may seem. As creatures of space and time, equipped with certain sensory organs, surely we have special ways of modeling our perceptions of color, for example; of categorizing and cross-correlating the entities whose localization in space and persistence in time renders them cognitively coherent (type taxonomies); and of analyzing spatial properties, such as parts structure, and the temporal behavior of these entities. To match our cognitive skills, AI systems will need analogous special methods.

The methods described here are designed to supplement a deductive question-answering algorithm that is now operational.† The algorithm draws on a base of logical propositions organized as a *semantic net*. The net permits selective access to the contents of individual

“mental worlds” and narratives, to sets of entities of any specified type, and to propositions involving any specified entity and classified under any specified topic. For example, if the story “Little Red Riding Hood” is inserted into the net (in logical form), the set of all propositions concerned with the wolf’s appearance can be separately and efficiently retrieved. In addition, certain property-inheritance mechanisms facilitate the transfer of information from generic entities and their parts (such as wolf) to particular entities and their parts (such as a real-world wolf).

In speaking of our system as a semantic net, we are not speaking from a particular camp. We believe that the issues addressed are bound to arise in any general knowledge representation sooner or later, whether it is based on semantic nets, frames, scripts, production systems, or anything else. These nominally disparate formalisms have much in common and appear to be converging further; for example, all incorporate a predicate-logic-like propositional language (however disguised in newspeak or enmeshed in code), all provide ways of clustering information so that the information brought to bear on a given task at a given time can be sharply limited, and all have (or are to be furnished with) property-inheritance mechanisms.

*A more detailed accounting of this work is available as technical report TR83-3 from the Department of Computing Science, University of Alberta, Edmonton, Alberta T6G 2H1.

†A report on this project headed by J. de Haan and L. Schubert is in preparation. Questions are at present posed logically, the English “front-end” being incomplete.

Recognizing type relationships

We believe that our net access organization and inheritance mechanisms do about as much as can be done to ease the computational burden of any *general* inference algorithm. By radically limiting the set of propositions allowed in the reasoning mill at any time, these mechanisms help to prevent the combinatorial explosions that are apt to bring the mill to a halt.

We need to do more than just limit the number of propositions to allow fast question answering, however. For, if all possible derivations of the answer to a question are long, then any general reasoning strategy will prob-

ably do a great deal of searching before finding one, even when working with a small set of propositions.

It turns out that standard deductive derivations of the answers to many simple questions are indeed rather long. Consider the question, "Whom does Mary love?" The desired answer is the set of persons Mary is known to love. Now suppose that the system finds, by retrieval of information about Mary under the topic "emotional attitudes" that Mary loves John *and* her prize orchid plant. We must confirm that John is a person (and therefore a suitable answer) while the orchid is not. The former sub-problem is not too taxing, assuming that the system has the facts, "John is a boy" and "Every boy is a person"

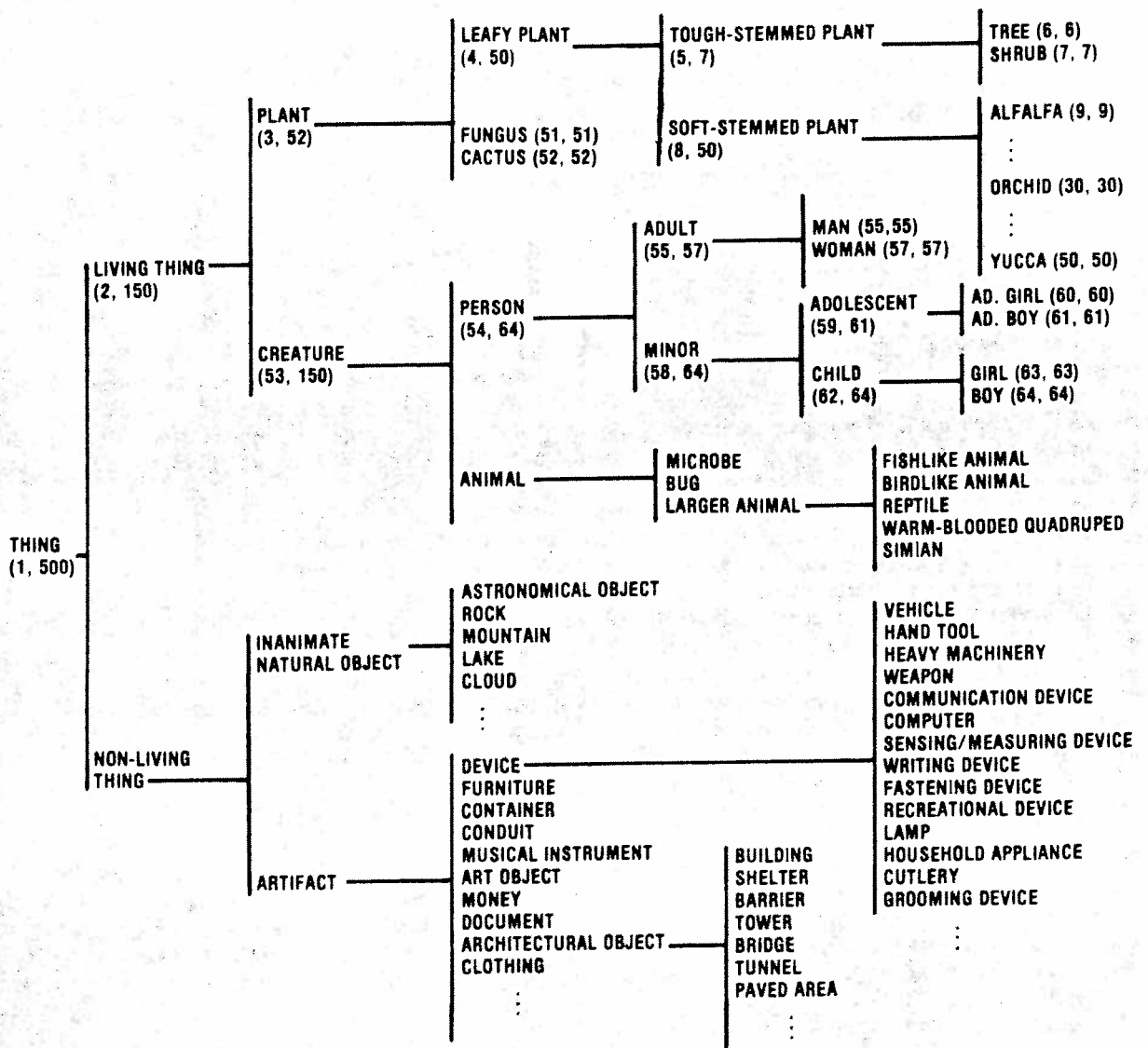


Figure 1. A partitioning hierarchy for physical objects. Each vertical bar corresponds to a partitioning of an object type into subtypes; e.g., [living-thing P plant creature]. Some of the number brackets based on preorder numbering of the hierarchy are shown. These allow the system to determine subordination and exclusion relationships in constant time; e.g., "orchid" is subordinate to "plant," since (30, 30) is included in (3, 52), and incompatible with "person," since (30, 30) and (54, 64) are disjoint.

at its disposal, and these are selected as relevant.* The latter subproblem, though, is harder than it ought to be. The following inference chain is required to solve it (where *o* is the beloved orchid):

- | | |
|--|---------------|
| (1) <i>o</i> is an orchid | known |
| (2) Every orchid is a soft-stemmed plant | known |
| (3) <i>o</i> is a soft-stemmed plant | from (1), (2) |
| (4) Every soft-stemmed plant is a plant | known |
| (5) <i>o</i> is a plant | from (3), (4) |
| (6) Every person is a creature | known |
| (7) No creature is a plant | known |
| (8) No person is a plant | from (6), (7) |
| (9) <i>o</i> is not a person | from (5), (8) |

This example is not the worst case; if, for example, Mary also loves her piano, more steps will be required to rule it out as a candidate answer, assuming that "creature" leads upward to "living thing" in the taxonomy of types (see Figure 1), while "piano" leads to "musical instrument" and hence to "artifact," and hence to "non-living thing," known to preclude "living thing." Yet subproblems such as establishing the non-personhood of an orchid or piano should not detain the reasoning system significantly.

In essence, we wish to be able to perform type compatibility checks for pairs of type concepts quickly. A pair of propositions such as "*o* is an orchid," "*o* is a person" should be directly *resolvable*, allowing proofs such as (1) through (9) to be short-circuited. Similarly, propositions that explicitly relate types to types, such as "Every orchid is a living thing," should be directly

evaluable to true or false, allowing their proofs or disproofs to be short-circuited. This dual need for resolving and evaluating propositions arises in all the special domains considered.

One method is based on finding intersecting upward paths in a type graph. This idea can be implemented as graph algorithms³ or as special theorem-proving strategies.⁴ Our own method, though closely related, entirely avoids path traversals and (under certain assumptions) determines type relationships in constant time.

The simplest version of the method can be understood by referring to Figure 1. The figure depicts a hierarchy of "P-assertions," each logically of the form $[T \text{ P } T_1 \dots T_k]$, meaning that type concept *T* is partitioned into the mutually incompatible, jointly exhaustive subtypes T_1, \dots, T_k . Each concept is labeled with a *number bracket* consisting of its preorder number and the highest preorder number among its descendants. If one node is an ancestor of another, its number bracket (regarded as an interval) contains that of the other. If neither is an ancestor of the other, the number brackets are disjoint. (A. Aho et al.⁵ discuss preorder traversal of trees in more detail.) These facts obviously allow compatibility and subordination checks in constant time.

*For those versed in predicate logic, our formal representation of these facts for input to the semantic net would be $[\text{John boy}]$ and $\forall x[[x \text{ boy}] \rightarrow [x \text{ person}]]$. This representation is based on an English-like infix form of predicate logic in which the predicate symbol, such as "boy" or "part-of," always follows its first argument (the subject) and is followed by the remaining arguments, if any. Throughout this article, when we refer to "facts," "propositions," or "assertions," we mean this type of representation.

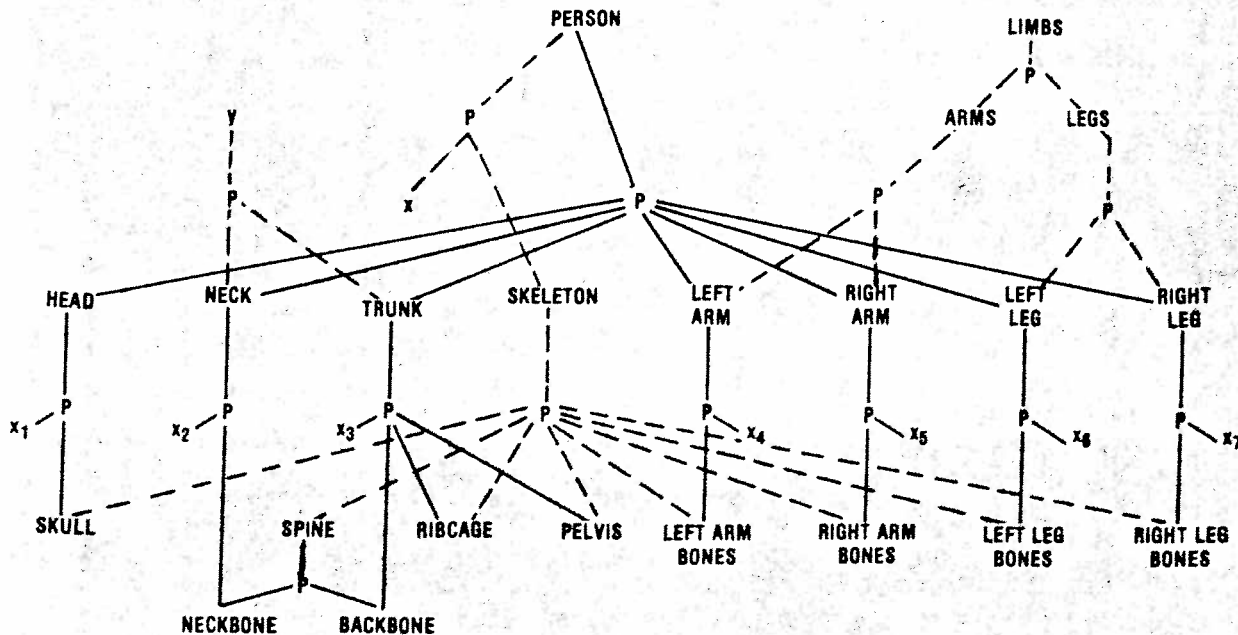


Figure 2. The upper levels of a partitioning graph for the human body. Each P-token represents a partitioning assertion dividing the node to which it is linked above into the nodes to which it is linked below. The solid lines define a partitioning hierarchy, and the broken lines define three additional, superimposed hierarchies.

The method can be generalized slightly. First, incomplete partitionings can be accommodated by using remainder categories to complete them. More significantly, *overlapping* hierarchies can be accommodated by equipping each concept with a separate number bracket for each hierarchy to which it belongs, along with a suitable hierarchy identifier. A compatibility or subordination check for two concepts now begins with a search for a common hierarchy identifier, and if one is found, proceeds to the comparison of number brackets. This check is still very fast, assuming that no concept participates in more than a few (say, two or three) hierarchies. Our attempts to draw up comprehensive concept taxonomies appear to support this assumption.

The generalized method, however, is logically incomplete, as we shall see in the following discussion of parts taxonomies.

Recognizing part-of relationships

The part-of structure of an object can be represented in essentially the same way as a taxonomy of concept types. We introduce an object partitioning relation P , with $[x P x_1, \dots, x_k]$ expressing that object x is (exhaustively) partitioned into parts x_1, \dots, x_k . If we simply want to assert that x has a part y , we can do so by writing $[x P y z]$, where z is a possibly empty remainder part.

Figure 2 shows a partial human anatomy, naively conceived, in the form of a P -graph. It consists of a main hierarchy and superimposed subsidiary hierarchies for the skeleton, the limbs, and the combination of neck and trunk y . (We are glossing over some logical niceties concerning the interpretation of graphs whose nodes are generic entities, such as "pelvis" and "left leg.")

The algorithm sketched for type graphs could be used here for detecting the incompatibility of such pairs of propositions as

x is John's pelvis; x is John's left leg,

or to determine the truth values of such propositions as

If x is John's skull and y is John's skeleton, then x is part of y ,

using number brackets based on preorder numbering of the overlaid hierarchies in the figure.

While useful, the algorithm is incomplete, as already noted. Consider the question, "Is the spine part of y ?" The response would be "unknown," since "spine" and y do not lie in a common hierarchy; yet the graph certainly allows the inference that the spine is part of y , since a P -assertion divides the spine exhaustively into neckbone (cervical vertebrae) and backbone, and these have upward paths to y . Similarly, we may ask whether the spine is part of the limbs, and again the response would be "unknown," since "spine" and "limbs" lie in no common hierarchy; yet a negative answer can be deduced from the graph.

The incompleteness of the methods just discussed has led us to seek more powerful methods. One method is complete and efficient for the class of "closed"

P -graphs, or P -graphs.⁶ The graph in Figure 2 can be closed by adding the (logically redundant) P -assertion $[x P x_1, x_2, \dots, x_n]$. A method for the still larger class of "semiclosed" P -graphs⁷ has been proved correct and complete.⁸

While closed and semiclosed P -graphs appear to provide much of the flexibility required for representing part-of (and type) structures, they still restrict us to certain kinds of overlaid hierarchies. Are there no efficient inference methods for unrestricted P -graphs, that is, for arbitrary sets of P -assertions?

Unfortunately, the chances of finding methods requiring no more than linear storage and time relative to the size of a P -graph are slim, since we would have to solve the famous unsolved problem " $P = NP$?" affirmatively. Thus, we will probably have to be content with incomplete special methods for type and parts taxonomies—a condition that is by no means disastrous, since the special methods are intended to supplement, not to replace, the general reasoning algorithm.

Recognizing color relationships

Imagine a witness to a bank robbery being questioned about the color of the get-away car. His impression was that the car was tan, and he is asked, "Was the car brown?" Clearly the answer should be "Yes, tan," and this answer could easily be deduced from the following (where c is the car in question):

- (1) c is tan
- (2) Everything tan is brown.

If the question had instead been "Was the car maroon?" a negative answer could have been inferred in four proof steps, from (1), (2), and the following:

- (3) Everything maroon is red.
- (4) Nothing red is brown.

These examples follow the pattern of the type and PART-OF inferences exactly and suggest that some sort of color hierarchy or graph should be used to eliminate searching. In fact, the 11 basic color terms of English could be introduced via the type partitioning,

- (5) [colored P red orange yellow green blue purple pink white black gray brown],*

and (2) could be reformulated as something like

- (6) [brown P tan rust midbrown chocolate. . .],

and similarly for (3), allowing either of the above questions to be answered by simple hierarchy methods.

However, a series of complications has led us away from graphical methods toward geometric methods. First, partitionings like (6) are inaccurate, since shades like tan, midbrown, and chocolate probably overlap. More accurate characterizations require partitioning these shades into overlapping and nonoverlapping parts.

*We regard it as a reasonable claim that every (uniform) color is at least a marginal instance of one of these basic colors, though for pragmatic reasons, people generally avoid applying color terms to their marginal cases.

Shades like turquoise and lime, which straddle boundaries between basic colors would also have to be subdivided, adding to the proliferation of partitionings.

Second, when we attempted to deal with "hedged" color relations, such as the statement that lime is sort of yellow and also sort of green, we realized that the color partitioning graph would at least have to be augmented with adjacent and/or nonadjacent relations. But even these additions would leave us totally unequipped to deal with other kinds of color properties and relationships such as lightness, purity, saturation, complementarity, and the warm/cool distinction. Geometric representations, on the other hand, offered a handle on all these problems. If colors could be represented as simple regions in some color space, all their properties and relationships could be read off their parametric representations.

With this objective in mind, we undertook a search for a suitable color space. Despite their commercial popularity, we rejected schemes based on mixing three primaries because they are theoretically incapable of representing all perceptually distinct shades of color⁹ and because the English color terms correspond to rather complex polyhedra in such schemes.

Our ultimate choice was a cylindrical representation (Figure 3), arrived at by imagining any color to be composed of some amount of a pure, monochromatic color plus certain amounts of black and white. Thus, one dimension runs through the continuum of rainbow hues, arranged in a circle, while the others parameterize the amount of black and white present (see figure caption).

This model, similar to models that are well-known to color theorists,¹⁰ covers the full range of perceptible shades. (Note that the saturation and lightness parameters used in these models do not coincide with the purity and dilution parameters of our model.) Our model appears to be unique in one respect because it renders each English color term simply as a region bounded by six coordinate surfaces (defined by three pairs of upper and lower parameter bounds). In all other models we are aware of, the boundaries are quite irregular. (Some adjustments to the simple regions in Figure 3 may be required. Perhaps part of the brown region separating red from black should be maroon or purple. We plan to investigate this empirically.)

With this color geometry, we can check any desired relationship between pairs of color regions, such as inclusion, overlapping, and adjacency, in a small, fixed number of comparisons. Moreover, it is easy to define nonbasic terms such as turquoise, maroon, beige, scarlet, and so on, as regions bounded (like basic color regions) by coordinate surfaces. Color properties such as lightness and purity and relations such as complementarity can be computed in fairly obvious ways.

We have tacitly assumed in the previous discussion that the color cylinder will contain explicit representations of all colors for which a compatibility check might ever be required. We can relax this assumption, allowing for the possibility that certain nonbasic colors are encoded only in terms of their qualitative relation to the basic colors. For example, turquoise might be axiomatized as being both sort of blue and sort of green, and lime as be-

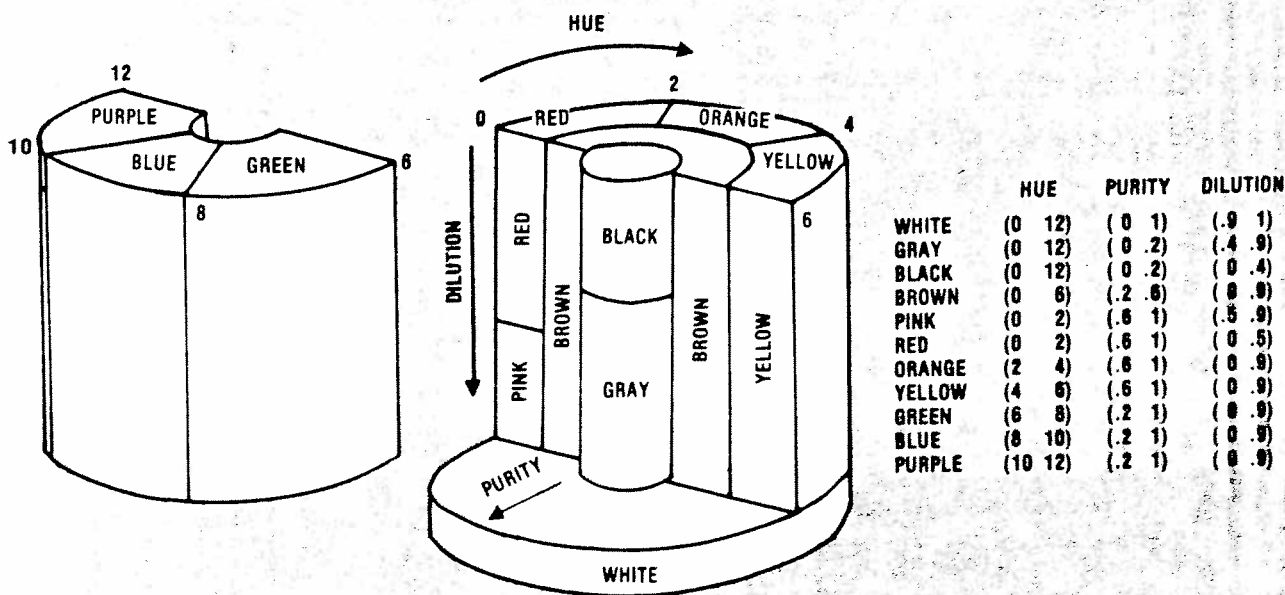


Figure 3. The 11 basic colors in a hue-purity-dilution color space with the cool shades "lifted away." Purity decreases as black is added to a pure color, and dilution increases as white is added to it. Purity = pure color/(pure color + black), and dilution = white/(pure color + black + white). The numerical values have been chosen on purely intuitive grounds. They could be quite drastically altered without affecting the results of the algorithms based on the model, as long as the region adjacency relationships are not changed.

ing both sort of green and sort of yellow. There is a method that allows the incompatibility of characterizations such as "sort of blue" and "sort of yellow" (and hence of turquoise and lime) to be detected by table lookup.^{8,11} The table also covers negated descriptions of this type, with and without the "sort of" qualifier.

Recognizing time relationships

Did the first moonwalk by an astronaut precede the first space shuttle launch? Most people can answer this question quickly and easily in the affirmative. The answer will perhaps be based on the feeling that the first moonwalk occurred prior to 1970, while the shuttle program became operational only in recent years. (More details than that may be recalled, of course.) Clearly, question-answering systems knowledgeable about events will likewise have to be able to store and recall approximate or exact event times.

The ability to retain absolute time information is not enough, however, since people easily recall the time order of connected sequences of events even without such information. Consider the fairy-tale, "Little Red Riding Hood," for example. Did LRRH meet anyone before arriving at her grandmother's cottage? The answer is "the wolf." In our system, the question-answering attempt would begin with associative retrieval of events that fit the pattern "LRRH encounters character x." Presumably, the events retrieved would include not only the first encounter with the wolf, but also the fateful second encounter, as well as the ultimate encounters with the gamekeeper and with the grandmother. Similarly, LRRH's arrival at the cottage would be retrieved associatively. The remaining task is then to sort out the prearrival encounters from the postarrival en-

counters, and for this task we need efficient methods for checking the time order of events.

Our representation for time encodes time intervals in terms of their end points, with a directed arc connecting each pair of time points whose order is known explicitly. The graph generated by a narrative is then an acyclic digraph (except for reentrant time travel stories). Any ordering relation implicit in it can be extracted by tracing from one point to the other, a linear-time operation relative to the number of edges of the graph.

As in the case of the other inference domains considered, however, we would prefer constant-time checks to linear searches. (In this respect, we would like to improve on heuristic methods such as those described by K. Kahn and G. Gorry.¹² Unfortunately, no methods are known for extracting ordering relationships from arbitrary acyclic digraphs in sublinear time without incurring nonlinear storage cost. (However, T. Kameda¹³ gives a constant-time method for certain restricted kinds of planar acyclic digraphs.) Rather than investing effort in this research problem (which would have limited pay-off in any case, since we would also like to introduce absolute times and durations or bounds thereon), we have proceeded pragmatically.

Roughly, the idea behind our scheme is to assign numeric values (pseudotimes, so to speak) to time points in their time order, when this order is known. To the extent that this assignment is possible, the time order of two time points can be checked in constant time by comparing their pseudotimes.

Figure 4 illustrates the kind of time graph determined by a narrative. Nodes denote time instants and are numbered in the order they are added to the graph. Also the pseudotimes (more or less arbitrarily incremented in steps of 1000 when not bounded above) are shown beside

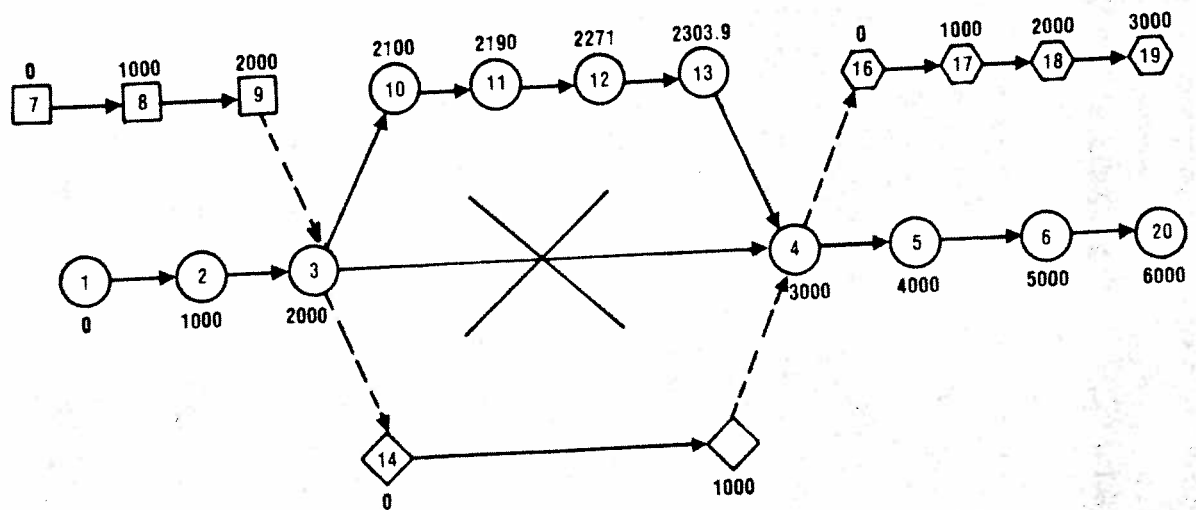


Figure 4. Time graph for a narrative. The numbers within the nodes record the narrative sequence, i.e., the order in which the nodes were added. The numbers beside the nodes are pseudotimes, used to check time order within a time chain. The four distinct node shapes distinguish the four time chains in the graph. The link from node 3 to node 4 (crossed off) is deleted when nodes 10-13 progress in intervals equal to 0.1 of the remaining pseudotime interval (i.e., 0.1 times the pseudotime at node 4 less the last assigned pseudotime)

the nodes. Typically, narrative events correspond to pairs of time nodes, such as 1, 2 and 3, 4). The graph consists of a collection of *time chains*, each with its own pseudotime sequence. In the figure, different node shapes are used for the different time chains; actually, this distinction is made by numerically typing each chain and maintaining a separate structure for interconnections between types (shown as broken links).

We can use the graph to check the time order of nodes 2 and 18, for example. After determining that 2 and 18 belong to different chains, the chain-to-chain (type-to-type) connections would be searched, yielding the cross-type link (4, 16) from the first to the second chain. Since the pseudotime of node 2 is less than that of node 4, and the pseudotime of node 16 is less than that of node 18, the answer "2 before 18" is returned.

Obviously time checks restricted to one chain require only one comparison; the worst-case computation time for time checks across chains is proportional to the number of chain-to-chain connections. This number is typically much smaller than the total number of links in the time graph, as far as we can tell from sample time graphs for newspaper stories several paragraphs long, a fairy-tale ("Little Red Riding Hood"), and excerpts from Hemingway's *The Old Man and the Sea* and from a book of European history.

An extension of the algorithm we are developing involves maintaining time bounds that delimit the *actual* time of each node and the actual duration of each link where such information is available. Maintaining such bounds by propagating upper bounds backward and lower bounds forward is a fairly simple matter.

The time graphs and associated algorithms provide a basis for the fast computation of a wide variety of temporal properties and event relationships, including time order, overlap, inclusion (during), and duration; exact or approximate time of occurrence; and exact or approximate elapsed time between events. All these are easily expressed in terms of the order of time points marking beginnings and ends of events, actual time bounds on these time points, and bounds on actual time intervals separating them.

We have shown that much combinatory reasoning in a question-answering system can be short-circuited by the use of special graphical and geometrical methods.

The domains considered—types, parts, colors, and times—do not quite exhaust those in need of special methods. In particular, part-of relationships are only one structural aspect of physical (and other) systems, and more powerful modeling methods are needed to rapidly infer static and dynamic relationships. For example, people intuitively sense the "faulty physics" in "He put a bunch of roses in the wine glass," perceiving with their "mind's eye" that the roses won't stay put (whereas violets might).

*Plenum—on the frontier of speech
and computer communication*

COMPUTER MODELS OF SPEECH USING FUZZY ALGORITHMS

by Renato De Mori, *Concordia University,
Montreal, Canada*

Dr. De Mori, a respected researcher and a professor of computer science, presents the results of research oriented toward the goal of making an experimental model capable of understanding spoken sentences of a natural language, uniting some of the best and most recent advances made in pattern recognition, artificial intelligence, signal processing, speech communication, logic, and information processing systems. Professor De Mori reports here on the conception of data-structures and algorithms for representing and using knowledge of speech analysis and perception in a speech understanding system. Beginning with a model for conceiving modules that perform tasks and combining them in a speech understanding system, the author proposes a system that is a flexible instrument for research rather than a prototype. Such a system can be updated whenever new convincing results appear in the scientific literature or are obtained by the system itself. This book shows how other approaches can be utilized in the framework proposed here and that most of the processes involved in speech understanding do not require precise numerical computations. Through a unified presentation, Professor De Mori explains how a flexible experimental model can be built using a fuzzy algebra for composing evidences. A volume in *Advanced Applications in Pattern Recognition*: Morton Nadler, general editor.

CONTENTS: Computer models for speech understanding. Generation and recognition of acoustic patterns. On the use of syntactic pattern recognition and fuzzy set theory. Design principles for controlling the use of structural rules for segmentation. Rules for characterizing sonorant sounds. Rules for characterizing the nonsonorant sounds. The lexical knowledge source. On the structure and use of task-dependent knowledge. Automatic learning of fuzzy relations. Towards a parallel system. Index.

508 pp., illus., 1983

\$59.50

ADVANCES IN CRYPTOLOGY

Proceedings of Crypto 82

edited by David Chaum, *University of California at Santa Barbara*, and Ronald L. Rivest and Alan T. Sherman, *Massachusetts Institute of Technology*

This volume provides a comprehensive, state-of-the-art view of cryptology, featuring the most significant cryptologic event of 1982: Adi Shamir's polynomial-time algorithm for breaking the basic Merkle-Hellman knapsack public-key cryptosystem. A unique record of the current state of cryptologic research, this is an invaluable source of information for everyone intrigued by the recent developments in this field. It is also well suited for use as a supplementary textbook in cryptology classes. The five main areas of study explored here are

- algorithms and theory
- applications
- modes of operation
- cryptoanalysis
- protocols and transaction security

348 pp., illus., 1983

\$45.00

Plenum
PUBLISHING CORPORATION

233 Spring Street
New York, N.Y. 10013

A good deal has been written on whether imagelike representations are psychologically real and theoretically necessary, but that is not at issue here. What *is* at issue is computational efficacy, and the methods of symbolic logic, though no doubt capable *in principle* of predicting the behavior of physical systems, need to be supplemented with special modeling methods to reach conclusions within reasonable times. The various expert systems incorporating models of toy blocks, electronic circuits, weight-and-pulley assemblies, and so forth will point the way, although the often complex and deformable objects of the real world (like plants, coats, and people) may require methods different from those of the popular microworlds. If sufficiently powerful analog models can be developed for physical objects, they may obviate the need for parts graphs, such as our *P*-graphs, just as the color cylinder obviated the need for color *P*-graphs.

We do not foresee having to devise many more special representations other than improved models for structural reasoning, as long as we are concerned with general question-answering and not with expert consultation (e.g., on programming, mathematics, or economic forecasting). Indeed, even specialized expertise may often require no more than redeploying spatiotemporal modeling skills. For example, expertise in symbol manipulation (as required for symbolic logic, mathematics, and programming) may well rest in part on spatiotemporal visualization and in part on linguistic skills (parsing, pattern matching), which are of course presupposed in a question-answering system. ■

Acknowledgments

Meticulous refereeing has helped us to condense and improve this article. The research reported was supported by operating grant A8818 of the Natural Sciences and Engineering Research Council of Canada.

References

1. L. K. Schubert, R. Goebel, and N. Cercone, "The Structure and Organization of a Semantic Net for Comprehension and Inference," *Associative Networks—The Representation and Use of Knowledge by Computers*, N. V. Findler, ed., Academic Press, New York, 1979, pp. 121-175.
2. A. R. Covington and L. K. Schubert, "Organization of Modally Embedded Propositions and of Dependent Concepts," *Proc. Third Biennial Conf. CSCSI/SCEIO*, 1980, pp. 87-94.
3. S. E. Fahlman, *A System for Representing and Using Real-World Knowledge*, MIT Press, Cambridge, Mass., 1979; see also the summary in *AI: An MIT Perspective*, Vol. 1, P. H. Winston and R. H. Brown, eds., MIT Press, Cambridge, Mass., pp. 453-470.
4. A. Bundy, L. Byrd, and C. Mellish, "Special-Purpose, But Domain-Independent, Inference Mechanisms," *Proc. 1982 European Conf. AI*, Orsay, France, pp. 67-74.
5. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *Data Structures and Algorithms*, Addison-Wesley, Reading, Mass., 1983.

6. L. K. Schubert, "Problems with Parts," *Proc. Sixth Int'l Joint Conf. AI*, Tokyo, 1979, pp. 778-784.
7. M. A. Papalaskaris and L. K. Schubert, "Parts Inference: Closed and Semi-closed Partitioning Graphs," *Proc. Seventh Int'l Joint Conf. AI*, 1981, pp. 304-309.
8. M. A. Papalaskaris, "Special-Purpose Inference Methods," masters thesis, Dept. of Computing Science, University of Alberta, Edmonton, 1982.
9. D. Judd and G. Wyszecki, *Color in Business, Science and Industry*, 2nd ed., John Wiley & Sons, New York, 1963.
10. *Ostwald. The Color Primer; Munsell. A Grammar of Color*, F. Birren, ed., Van Nostrand Reinhold, New York, 1969.
11. M. A. Papalaskaris and L. K. Schubert, "Inference, Incompatible Predicates and Colours," *Proc. Fourth Biennial Conf. CSCSI/SCEIO*, 1982, pp. 97-102.
12. K. Kahn and G. A. Gorry, "Mechanizing Temporal Knowledge," *Artificial Intelligence*, Vol. 9, 1977, pp. 87-108.
13. T. Kameda, "On the Vector Representation of the Reachability in Planar Directed Graphs," *Information Processing Letters* 3, 1975, pp. 75-77.



Lenhart K. Schubert is an associate professor of computing science at the University of Alberta, Edmonton, with research interests in various aspects of artificial intelligence, especially knowledge representation and natural-language understanding. He received a PhD in aerospace studies from the University of Toronto in 1970 and conducted postdoctoral studies at The Johns Hopkins University in Baltimore and at the University of Alberta under a National Research Council of Canada postdoctoral fellowship. Schubert is a member of the ACM, the Association for Computational Linguistics, the Cognitive Science Society, the Canadian Society for Computational Studies of Intelligence, and the American Association for Artificial Intelligence. He is a former Alexander von Humboldt research fellow.



Mary Angela Papalaskaris is a doctoral student in the Department of Artificial Intelligence at the University of Edinburgh. Her research interests center around formal methods in artificial intelligence, particularly automated reasoning. She received a BSc in computer science from Lakehead University, Thunder Bay, Ontario, in 1979 and an MSc in computing science from the University of Alberta, Edmonton, Alberta, in 1982. Papalaskaris is a member of the ACM.



Jay Taugher is a graduate student at the University of Alberta, Edmonton, where he is pursuing an MSc in computing science. His research focuses on knowledge representation in general and temporal knowledge in particular. A former employee of Burroughs Corporation, his previous work involved relational database systems and secure operating systems. Taugher received a BSc in computing science from Queen's University, Kingston, Ontario, in 1981, and is a member of the ACM.

The authors' address is University of Alberta, Edmonton, Alberta, Canada.