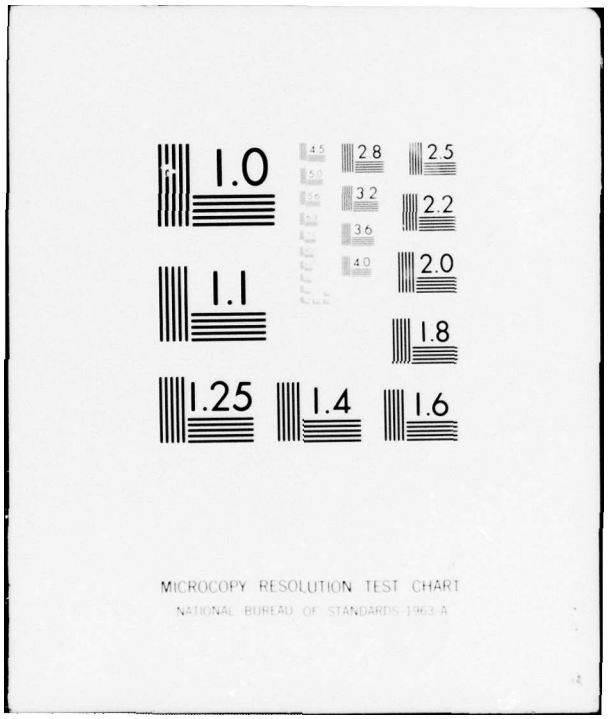
AD-A040 883 RAND CORP SANTA MONICA CALIF F/G 6/4
THE ROLE OF PARTIAL AND BEST MATCHES IN KNOWLEDGE SYSTEMS,(U)
JAN 77 F HAYES-ROTH
UNCLASSIFIED P-5802 NL

1 OF 1
AD A040883

END
DATE FILMED
7-77


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0
4.5
5.0
5.6
MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

(2)

(6) THE ROLE OF PARTIAL AND BEST MATCHES IN KNOWLEDGE SYSTEMS,

(10) Frederick Hayes-Roth

(11) January 1977 (12) 39p.

(14) P-5802

296600

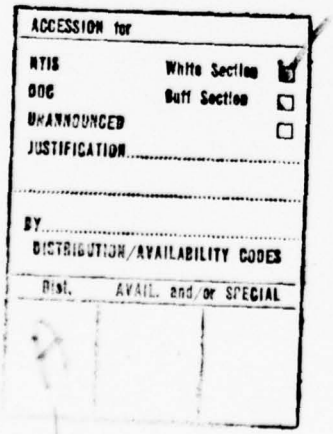
The Rand Paper Series

Papers are issued by The Rand Corporation as a service to its professional staff. Their purpose is to facilitate the exchange of ideas among those who share the author's research interests; Papers are not reports prepared in fulfillment of Rand's contracts or grants. Views expressed in a Paper are the author's own, and are not necessarily shared by Rand or its research sponsors.

The Rand Corporation
Santa Monica, California 90406

ABSTRACT

Partial matching is a comparison of two or more descriptions that identifies their similarities. Determining which of several descriptions is most similar to one description of interest is called the best match problem. Partial and best matches underlie several knowledge system functions, including: analogical reasoning, inductive inference, predicate discovery, pattern-directed inference, semantic interpretation, and speech and image understanding. Because partial matching is both combinatorial and ill-structured, admissible algorithms are elusive. Economical solutions require very effective use of constraints that, apparently, can be provided only by globally organized knowledge bases. Examples of such organizations are provided, and promising avenues of research are proposed.



INTRODUCTION: WHAT IS THE PARTIAL MATCHING PROBLEM?

A partial match is a comparison of two or more descriptions that identifies their similarities. Because typical descriptions comprise symbolic property-lists or propositional formulae, a partial match of two descriptions includes three components: an abstraction, consisting of all properties or propositions common to both compared descriptions; and two residual terms, representing the properties that are true of only one or the other of the descriptions. If the two compared descriptions are A and B, the partial match of A and B, denoted PM(A,B), is (A*B, A-A*B, B-A*B), where A*B denotes the abstraction of A and B, and A-A*B and B-A*B denote the properties of A and B, respectively, that are not contained in A*B. In other papers, partial matching has been variously referred to as interference matching, generalization or correspondence mapping [9, 10, 14, 15, 37, 40].

The premise of this paper is that the partial matching problem is of fundamental importance for pattern-directed inference and other knowledge-based activities. While some well-structured problems may be solvable by conventional algorithmic methods, it appears that the majority of complex problems cannot be solved with a small set of predefined, pattern-matching rules that are applied in an all-or-none fashion, exactly as coded. Just as laws must be flexibly interpreted to regulate complex social interactions in reasonable ways, so is it true in systems employing large amounts of

knowledge to complex problems that each element of knowledge should *influence* the outcomes of numerous decisions without *dominating* any. In such systems, many diverse sources of influence must be pooled to identify the *best* or *most strongly indicated* course of action at each moment in time. Partial matching and best matching are the mechanisms for accomplishing this control.

In addition to its role in identifying the commonalities and differences of comparable situations, partial matching can be interpreted in two other ways. The second role of partial matching is to ascertain how well an observed event satisfies the prescribed constraints of an *ideal* or *prototypic* situational description. Identifying the best match between the description of an observed event and alternative prototypes enables the current situation to be *recognized* as an *instance* or special case of one of the prototypes. Those relationships shared by both descriptions are the *constraints* of the prototype that the observed event satisifies. Any residual properties of the prototype are unsatisfied constraints. Classifying an event according to its best match among alternative prototypes is tantamount to pattern recognition by constraint satisfaction (Cf. [1]).

The third role of partial matching is similar to constraint satisfaction. In this case, too, a description of data is compared with descriptions called *templates*, *case frames*,

schemata or frames. These frames are usually hierarchically organized, empirical or conceptual descriptions of observable phenomena. In short, frames constitute a system's knowledge of its world. When the best matching frames are ascertained, the data are interpreted by imposing the frame structure upon them. For example, in a speech understanding task the data might consist of an array of hypthesized words, and the templates would be empirical phrase structures of the language. The best-matched templates determine how the words should be parsed and semantically interpreted. As a general rule, it appears that semantic interpretation is best conceived as the mapping between current data and previously inferred schemata. Because the superficial aspects of most observed situations differ substantially from all previously encountered ones, semantic interpretation is fundamentally a problem of partial matching.

In the next section, several applications of partial and best matches are presented to convey the generality and difficulty of the partial matching problem. Subsequently, a criterion for the admissibility of partial matching algorithms is discussed which, though simple and reasonable, is difficult to realize. In the last sections, the principal features of the partial matching problem are discussed, and some promising approaches toward its solution are proposed.

SOME APPLICATIONS OF PARTIAL MATCHING

In this section, several applications are briefly discussed to illustrate the generality, importance, and difficulty of the partial and best match problems. The applications considered include analogical reasoning, semantic interpretation, inductive inference, predicate discovery, pattern-directed inference, and speech and image understanding. In each case, the central problem is finding a best match between two data descriptions or between a data description and existing knowledge. This nearly always entails searches of exponential problem spaces.

Analogical Reasoning. While this category properly embraces numerous problems of widely varying specificity, the most well studied is "A is to B as C is to which, D1, D2, ..., Dn?" As several researchers have shown [6, 38], an effective program for solving these problems is as follows:

(1) Compute the partial matches PM(A, B), PM(C, D1), ..., PM(C, Dn).

(2) Determine the best match between PM(A, B) and one of PM(C, D1), ..., PM(C, Dn). If the best match is PM(C, Dk), Dk is the best solution to the problem.

Recall that PM(X, Y) comprises three terms, the abstraction X*Y and the residuals of X and Y. Thus, the partial match between A and B defines a viewpoint for interpreting what changes were necessary to transform A into B; i.e., the pair A-B induces a transformation [A -> B]. This transformation is implicit in the structure PM(A, B) = (A*B, A-A*B, B-A*B): A*B specifies which properties of A were retained, A-A*B specifies which properties

of A were deleted, and B-A*B specifies which properties were added to A by the transformation of A into B.

The partial match between PM(A, B) and PM(C, Di) (for some i) can be viewed as a comparison of two ordered lists and is defined as PM(PM(A, B), PM(C, Di)) = (((A*B)*(C*Di), (A-A*B)*(C-C*Di), (B-A*B)*(Di-C*Di)), R1, R2), where R1 and R2 are the appropriate residual terms. The abstraction of this partial match consists of three terms: (A*B)*(C*Di) comprises all properties common to all of the descriptions, A, B, C, and Di (the partial matching operator * is associative); (A-A*B)*(C-C*Di) comprises all properties removed from A and C in transforming them to B and Di, respectively; and, similarly, (B-A*B)*(Di-C*Di) comprises all properties added to A and C in transforming them to B and Di, respectively. Thus, the original analogy problem is reducible, through partial matching, to a question of choosing the one combination of common, deleted, and added properties that is most persuasive or plausible. Because any answer to this question must rest on empirical or subjective criteria, nothing of general validity can be added to this analysis.

Another use of partial matching for analogical reasoning occurs in Merlin [28]. In this system, any object can be interpreted as a special case of another whenever their differences do not outweigh their similarities. As an example, suppose we wished to play baseball with only a bat and a tennis

ball. In Merlin's framework, the feasibility of playing should be directly related to the reasonability of viewing a tennis ball in the role of a baseball. Such a viewpoint can be achieved by partial-matching their descriptions. Suppose tennis ball were defined as a "bouncy, hollow, light, fuzzy, four-inch spheroid that is forcefully hit in the game of tennis" and a baseball were defined as a "hard, solid, leather-covered, moderately heavy, four-inch spheroid that is forcefully hit in the game of baseball." In this case, the abstraction of the two descriptions specifies that both objects are four-inch spheroids hit forcefully in games. The residuals, however, specify that whereas the baseball is hard, solid, leather-covered, moderately heavy and used in the game of baseball, the tennis ball is bouncy, hollow, light, fuzzy and used in the game of tennis.

To decide if the tennis ball will suffice as a makeshift baseball, these residuals must be reconciled. One simplifying approach to reconciliation employs semantic categories. If correspondences between pairs of residual properties can be established so that each difference is interpretable as a specific dimensional variation, the significance of the overall difference can be decomposed and, thus, easily apprehended and evaluated. A hierarchical organization of the system's knowledge greatly facilitates such a decomposition. For example, the difference hollow-solid can be reconciled by interpreting it as a variation on the dimension of "structure" or "construction type." As a result, a tennis ball can be viewed as a type of baseball

that is hollow (rather than solid), light (rather than moderately heavy), fuzzy (rather than leather-covered), used in the game of tennis (rather than baseball), and bouncy (rather than some unspecified related property of a baseball). If these differences do not outweigh the similarities of the two, the tennis ball will serve admirably.

Before leaving this example, consider the role of partial matching and residuals in establishing the correspondence between objects. First, the two objects' descriptions were obtained from a dictionary or semantic network. Second, the properties common to both were abstracted by intersecting their property-lists. Third, the residuals were forced into possible corresponding value pairs by finding dimensions that embraced both values. Note that, in general, reconciling the difference between two arbitrary values requires a recursive application of the partial matching scheme. Finally, the best match maximizes the similarities and minimizes the differences (according to exogenous criteria) between the compared descriptions.

Other sorts of analogical reasoning tasks can be formulated easily. For example: (1) If I know a detailed procedure (ordered operations on operands) to accomplish a specific function (establish particular relationships on the operands), how do I modify the procedure to accomplish similar objectives on qualitatively different operands? Answer: try to find related operations applicable to the new operands that perform similar

functions. (2) If I want to persuade someone that X causes Y but don't have specific examples, what can I do? Answer: find an example where X' caused Y' and X is to X' as Y is to Y'. Despite the fact that such arguments are not strictly logical, many people find them persuasive when the underlying analogies are plausible.

Semantic Interpretation. The assignment of best-matched frames as the semantic interpretation of verbal material was previously mentioned. There is a second way in which partial matching supports semantic interpretation. In this case, two or more concepts sharing certain syntactic relationships stimulate restricted sorts of "spreading activation" searches of a semantic network. When the searches emanating from the original concepts intersect, the connecting path defines the semantic interpretation of the syntactic structure [24, 31]. For example, a novel noun-noun phrase encountered in a text, such as "lawn mower," can be semantically interpreted by finding the best match among the relationships that radiate from the two concepts "lawn" and "mower" in a network embodying dictionary definitions. In this example, the best such match entails the following paraphrased interpretation: a "lawn mower" is a machine that cuts grass or similar plants [24]. Spreading activation, intersection searches are now widely applied in computer science and psychology. Their similarity to the search techniques employed by Merlin is apparent. Regardless of the particular knowledge representation adopted, the essential function of these systems

is to find the best match possible under the constraints imposed by the current knowledge.

Inductive Inference. Several researchers have shown that patterns, concepts, and production rules can be inferred by partial-matching examples to discover the consistently repeated, hence presumably criterial, properties [3, 4, 8, 9, 10, 14, 15, 18, 19, 35, 37, 40]. To illustrate, consider the following examples of several classes:

> Example 1: Tom and Jack are brothers. Jack is the father of a boy named Bill who is under 10. Both Tom and Jack are in their fifties. Jack's brother is Bill's Uncle Tom.
>
> Example 2: Mary is the mother of twin sons, Bill and Jim. Mary is in her forties, while the boys are both 14. Mary has two brothers who are the boys Uncles Tom and Steve.
>
> Example 3: Sue has no brothers or sisters. Her mother is Jane, and Jane has has a brother named Fred. Fred is Sue's uncle.
>
> Example 4: Fred was a brilliant Negro who lived all of his life in a predominantly white, racist country. Because he was powerless and intimidated, Fred was humiliatingly subservient to the whites in his community. Fred was an Uncle Tom.
>
> Example 5: Because John, an aging, impoverished Negro, was humiliatingly subservient to Southern whites, the young blacks in his town called him Uncle Tom.

These examples will support a number of both correct and incorrect inferences that are equally plausible. For example, if Examples 1 and 2 are partial-matched, one inference is that parents are at least 40 years old and children are 14 or younger. However, the type of inference that I want to draw attention to

here has to do with notions of "Uncle." By partial-matching Examples 1 and 2, it is reasonable to infer that an uncle of x is the brother of the parent of x. However, the best partial match of these two examples would entail the stronger inference that x's Uncle Tom is the brother of x's parent, who is at least forty, while x is no older than 14.

A valid inference of the concept of "uncle" requires partial-matching all of Examples 1, 2 and 3, whereas a valid inference of the concept of "Uncle Tom" requires comparing Examples 4 and 5. This illustrates one of the perplexing problems regarding the role of partial matching in inductive inference. While it is possible to infer valid rules by partial-matching enough examples to eliminate all irrelevant properties, partial matching is also necessary to determine which examples illustrate the same concept. Knowing that Examples 4 and 5 should be compared to infer the meaning of "Uncle Tom," rather than comparing Examples 1, 2, 4, and 5, requires additional knowledge.

Suppose a learning system were asked to decide, based only on its knowledge of the five examples, if a certain 55-year-old Negro named Sam could be considered an uncle. To answer, it would necessarily seek similarities between the properties of Sam and previous examples of uncles. If, instead of actually retaining all examples, the system had only stored some "sufficient" set of rules induced by partial-matching arbitrarily

selected subsets of examples, its current classification would have a good chance of being incorrect. Because most systems do, in fact, attempt to store only a minimal set of rules that can "cover" the data [25, 35], they are prone to errors caused by decisions, about what combinations of properties are important, made before the properties of a test item are known. A system that stores its examples and postpones inferencing until the item to be classified is fully specified has a significantly reduced probability of error. In the current example, such a system would be guaranteed to have sufficient evidence to infer both that: if Sam is the brother of a parent, he may be labeled an uncle; and if he is subservient to whites, he may be an Uncle Tom.

The important point to observe is that the properties of the item to be classified, not the properties of the training data, determine which inferences should be made. Obviously, then, many inferences cannot be anticipated or generated until the problem is fully specified. In short, optimal performance in inductive inference requires a "wait-and-see" approach. In actual applications of the partial matching mechanism to pattern classification, the improved performance of wait-and-see classifiers has repeatedly been observed [5, 11].

The general learning framework that revolves about partial matching has been applied to the induction of several kinds of knowledge, including speech and image patterns [5, 9, 11, 35],

structured or relational concepts [3, 9, 10, 14, 15, 37, 38, 40], transformational grammar rules [9, 10, 38], and other [condition -> action] productions [38].

Predicate Discovery. While the type of induction discussed in the previous section assumes the prior discovery and encoding of those properties needed to express a rule, partial matching provides a basis for discovering new predicates too. For example, if a learner were exposed to the following sentences, it would have a good basis for several interesting inductions:

> Example 1: Because John is so tall, it is difficult to find clothes that fit him.
>
> Example 2: Because Mary is so short, it is hard to get clothes that can fit her.
>
> Example 3: Because Joanne is so fat, it is impossible to get apparel that is the right size.
>
> Example 4: Because Tom is so skinny, it is not possible to find clothes that are suitable.

Using only superficial characteristics of the string representations of these examples, the following common abstraction would be produced by partial-matching:

(Because u is so v, it is w to x).

The corresponding residual values from the four examples associated with each variable u, v, w and x are as follows:

u: (John, Mary, Joanne, Tom)

v: (tall, short, fat, skinny)

w: (difficult, hard, impossible, not possible)

x: (find clothes that fit him,
get clothes that can fit her,
get apparel that is the right size,
find clothes that are suitable).

Thus, with only four examples and very little knowledge, reasonable inferences regarding four apparent categories of natural language could be generated. The four distinct values associated with each of the variables are apparently subsets of the possible domains of associated (unknown) predicates. For example, John, Mary, Joanne and Tom are four of the possible values of the attribute "name." If this attribute had already been known to the system, partial-matching of the examples would have preserved the common "name" attribute, and a slightly more informative abstraction would have been produced, such as:

(Because the thing named u is so v, it is w to x

Thus, u, v, and w contribute to the discovery of the categories of name, body shape attributes, and expressions for "difficult to achieve". For the purposes of machine learning, knowledge of these interpretations per se is unnecessary. All that apparently is necessary is to infer the existence and composition of such categories (unary predicates), and this may be done whenever different constants are correspondents in correctly partial-matched descriptions.

Continuing with the previous example, it is also interesting to compare the residuals associated with variable x by a

recursive application of partial matching like that employed in Merlin. As a result of recursive partial matches of the four residual x strings, the following sequence of inferences will be produced:

(1) Infer the category FIND = {find, get}.

(2) Infer the category CLOTHES = {clothes, apparel}.

(3) Infer the category FIT = {fit him, can fit her, is the right size, are suitable}.

Then the abstraction of the residuals of x is:

(FIND(a) CLOTHES(b) that FIT(c)).

Notice that this abstraction is itself a candidate for a new type of ternary relation that, by definition, is true of any triple (a, b, c) constituted from the categories FIND, CLOTHES, and FIT, respectively. Any such triple is an instance of this general template and has the obvious interpretation. Such a template is a plausible model of the natural language expression for finding clothes that fit. In any case, a capacity exists to identify plausible syntactic categories and semantic templates by partial-matching even a small number of similar verbal strings. This approach to predicate discovery has been successfully applied to a number of restricted languages [9, 17, 36].

Pattern-directed inference. One of the concepts that has captured the imagination of many computer scientists and psychologists is that of frames, prototypes, templates, scripts

or schemata [2, 26]. Frames are supposedly knowledge units that delineate the elements of physical or conceptual events and express the constraints by which they are related. Distinct frames have been proposed for every ordinary physical object, typical configurations of objects, and most observable phenomena (e.g., dining at a restaurant or shopping for food). While there is *prima facie* evidence supporting the theory that people have such knowledge, there is little concrete understanding of how this knowledge can be exploited to simplify reasoning processes. What can be universally agreed upon is trivial: whenever a situation is encountered where existing knowledge is applicable, that knowledge should be applied to constrain the possible interpretations attributed to observed phenomena.

In this framework, the key issues are how relevant knowledge can be identified efficiently and applied effectively. Thus, for the moment, it will be assumed that a frame exists for describing every interesting pattern of relationships. Suppose, for example, that the number of frames relevant to image processing is about 100,000, including ones for familiar faces, buildings, automobiles, buses, bodies, trees, mountains, furniture, and implements of various sorts. Now, suppose that someone presents a photograph selected randomly from a magazine and asks how knowledge should be employed to assist in interpreting it. Simply asserting that we should apply whatever knowledge is needed to resolve the a priori uncertainty about the identity of various objects and their interrelationships is not an answer,

for this is presumed by the question. The question asks how the relevant knowledge can be identified. Once again, the answer appears to be that the best-matching frames should be chosen to interpret the data. In most cases, even best-matched frames will only be partially satisfied, because observed objects are occluded or otherwise fail to conform perfectly to the preconceived frame constraints. Once the best-matched frames have been identified, their knowledge can be exploited to hypothesize and test the apparently missing or erroneous data constituents.

Because no frame, by itself, can be expected to give a thorough account of the significant features of any normal, reasonably complex scene, satisfactory interpretations will normally require the integration of several partially matched frames. Two ways of determining the appropriate combination of frames can be proposed: (1) frames should be tried one-at-a-time, and additional frames should be incorporated as needed to resolve residual or anomalous properties; (2) some identifying characteristics of appropriate frames should be discerned through an analysis of global properties of the problem, and then frames satisfying these dynamically determined criteria should be invoked. In the next subsection some recent results of speech and image understanding research are presented favoring the second alternative.

Speech and Image Understanding. Speech understanding

systems face the task of finding the best-fitting interpretation for a noisy, parametric time series. The parameters are acoustic measurements and the interpretation is a hierarchical tree whose root is a semantic template from the language and whose intermediate levels represent phrases, words, syllables, phones, and acoustic segments [16, 20]. An interpretation is constructed by applying knowledge of possible mappings between intermediate levels. In the Hearsay-II system in particular, the interpretation process occurs basically in two phases. First, knowledge about the acoustic realization of words is used to hypothesize, bottom-up, plausible words at various temporal locations within an utterance. For example, if the sentence contains 10 words chosen from a 1000-word vocabulary, about 7 or 8 on the average are correctly hypothesized. In addition, approximately 200 incorrect words are hypothesized, and about 40 of these are actually rated higher than valid word hypotheses.

In the second phase, missing words are hypothesized and rated and the entire sequence of words in the sentence is parsed and assigned an overall semantic interpretation. The key problem in this phase is to generate and rate the most plausible, missing words. Even when the vocabulary and grammar are highly constrained, the size of the search space for possible grammatical word sequences is extraordinarily large. In the Hearsay-II system several approaches to this problem were tried, and only one approach apparently derived sufficient constraint, by applying enough knowledge simultaneously, to succeed. The

method used was to partial-match the entire collection of bottom-up word hypotheses against all templates of the grammar, in parallel, in the hope of finding one sequence of highly-rated words that was grammatical and most probably valid. If such a sequence could be identified, the system predicted and rated its plausible word extensions, iteratively, until a complete interpretation of the sentence was constructed.

Two knowledge sources were involved in computing the partial match between the matrix of hypothesized words and the grammatical case frames. These were WOSEQ [21], a word sequence hypothesizer, and PPARSE [12], a partial parser. In overview, WOSEQ uses knowledge about the adjacency of words in the language to form hypothetical word sequences by concatenating successive language-adjacent and time-adjacent word hypotheses. It prunes the search space further by terminating the concatenation process for any sequence when the expected benefit is less than the cost, i.e., when the increase in credibility obtainable by concatenating additional word hypotheses is insufficient to warrant the attendant multiplicative increase in the total number of word sequences generated. Each of the most credible word sequences identified by WOSEQ is then evaluated by PPARSE to determine whether it is actually grammatical, i.e., whether it is a subsequence of some sentence in the language. Each of these partial matching procedures is now explained in more detail.

WOSEQ uses a precomputed bit matrix that specifies for each

possible word pair (u, v) whether the sequence u v can occur in a sentence of the language. For the 1000-word vocabulary, this requires approximately 30K 36-bit words of memory. Given a collection of bottom-up word hypotheses, WOSEQ selects a few of the most credible ones as seeds for its sequence-growing process. Each seed is a one-word sequence, and the following procedure is applied repeatedly to all sequences until quiescence occurs:

> (1) For each word sequence W, construct the sets P(W) and S(w) of word hypotheses that can precede and succeed W. P(W) contains all hypotheses that are both language-adjacent and time-adjacent to the first word in W. The set S(W) contains all hypotheses that are time and language-adjacent to the last word of W.
>
> (2) For each w in P(W) evaluate the credibility of the sequence (w, W). This is an increasing function of the credibility of w and W, an increasing function of the total number of syllables spanned by (w, W), and a decreasing function of the number of words in P(W). If the credibility of the sequence (w, W) is greater than that of W, add (w, W) to the set of hypothesized sequences. For each word w in S(W), similarly process the potential sequence (W, w).

When WOSEQ quiesces, it will have identified sequences of pairwise-grammatical words that appear to be most credible over the entire set, both because they incorporate at least one of the individually most credible bottom-up hypotheses and because they satisfy a maximum number of low probability constraints. WOSEQ is usually successful at its task, because it continually increases the credibility of the objects it processes. It does this by adducing contextual support in the form of numerous, consistent, unlikely observations. The algorithm is efficient because the time and language-adjacency constraints are easily

computed. In a later section of this paper, it is suggested that easily computable, global attributes of the problem space may provide a promising, general approach to the partial matching problem.

The next step in the linguistic partial matching problem is to test each word sequence for grammaticality. This requires a parser capable of recognizing the grammaticality of any word sequence, even if it is only a subsequence of the string derivable from a nonterminal. In Hearsay-II, this is accomplished by a program PPARSE. PPARSE is a bottom-up, left-to-right Kay-type parser with the following modifications: Any rewrite rule such as X -> A B can be applied, and the parse node X constructed, whenever the leftmost derivative of B in the parse tree is the first word of the sequence being partial-parsed. Similarly, any rewrite like Y -> C D can be applied whenever the rightmost derivative of C is the last word of the sequence being partial-parsed. These are the only cases in which incomplete tree structures are built.

WOSEQ and PPARSE succeeded at controlling the combinatorics of the search problem, while a number of production systems failed [16, 27], because hypotheses that satisfy many of WOSEQ's constraints are likely to be valid. Furthermore, the truly expensive operation in this partial matching, instantiating and hypothesizing incomplete grammatical case frames, occurs only when an incomplete nonterminal can appropriately derive the first

or last word of a sequence selected by WOSEQ. Compared to any simplistic conception of how a frame system can operate to hypothesize and then fill in partially instantiated frames, WOSEQ and PPARSE constitute a significantly superior solution to the best match problem.

The last example of partial matching to be considered is the problem of determining stereo disparity between two images that are left and right-eye views of one scene. To resolve the disparity between two images of this sort, it is necessary to partial-match them to identify the corresponding (same) objects in each image. Once this is done, the lateral displacement or disparity between the two is a cue for the distance of the object from the viewer. The human visual system is capable of resolving such disparity, even when there are no distinguishable objects in either view (as in random-dot stereograms). Recently Marr and Poggio [22] have shown how the necessary partial matching computations can be performed locally by spatially distributed, cooperative processes. Their approach rests on the observation that, while the disparity between any two corresponding points is initially unknown, any hypothesis regarding some particular disparity value between two points in the two images implies approximately the same disparity value between neighboring points. By constructing a problem representation in which every possible pair of corresponding points, with disparity d, influences the neighboring points with matching properties toward correspondences under the same disparity, a difference equation

is constructed that can be applied iteratively and locally to choose correspondences that maximize constraint satisfaction. A solution in this algorithm is just a steady-state reached by the difference equation.

This application of partial matching is particularly interesting, because it shows how global features of the problem space, such as disparity and spatial position, can constrain the search for the best match. The global communication of constraint is accomplished by directly connecting neighboring points whose hypothetical disparity values influence one another. To develop a mechanism capable of this sort of information sharing, a representation had to be discovered that clarified the relationship between global data attributes (location and disparity) and local computations involved in partial matching (determining the grey-scale similarity of two potentially corresponding points). The role of this integrated global-local problem representation is comparable to that played by the precomputed language-adjacency matrix used by WOSEQ to hypothesize word sequences in Hearsay-II. This suggests some interesting properties of the partial matching problem that are pursued in the subsequent sections.

PRINCIPAL PROPERTIES OF THE PARTIAL MATCHING PROBLEM

From the preceding illustrations, it is possible to identify four principal characteristics of the partial matching problem.

In this section, these are briefly discussed.

The desirability of analyzing any particular configuration of data can only be determined dynamically. In the large class of problems where partial matching is necessary and computationally expensive, the number of distinct partial matches that can arise is virtually limitless. As a result, it is not possible to predetermine all combinations of observable properties that may, at some time, most warrant some response. A fortiori, it is not possible to rank order the potential situations in terms of import or interest value. Rather, the choice of which configurations of data deserve further processing resources is determinable only as a result of dynamic partial matching between the data in hand and the frames or templates specifying known constraints.

Partial matching, as a general computational problem, is intractable. Because partial matching subsumes the graph monomorphism, the k-clique, and other NP-complete problems, the amount of time apparently needed to solve worst-case problems is at least exponential in the complexity of the structures being matched. It follows that if partial-matching is to be applied successfully, problem complexity must be reduced. The principal way in which such complexity reduction can be accomplished is by choosing rich, high-order predicates as a basis for description. As the grain of description is reduced toward uniform, low-level predicates (e.g., simple graphs, retinal arrays of on-off

detectors, semantic primitives), the partial matching problem is made inherently more complex and less feasible.

Partial matching is fundamentally nondeterministic. Thus far in this paper the nondeterminism of partial matching algorithms has been neglected, primarily because one partial match solution is usually best. Thus, while any program designed for partial matching must incorporate logic that permits it to pursue multiple solutions simultaneously, effective mechanisms will quickly prune poor alternatives from consideration.

Good partial matches traverse a priori boundaries and multiple levels of hierarchically organized knowledge structures. This point is of the utmost importance for understanding why simple approaches to pattern-directed inference or frame-theoretic analysis of real data are likely to fail. Simple approaches will attempt to hypothesize all partial-matched frames and then predict and verify their missing constituents. In any reasonably complex domain, the best interpretation of data will traverse a priori boundaries of several low-order frames and will only be apparent when multiple levels of partial-matched frames are integrated. The simple approach entails extensive unwarranted searching of many levels of frames, because hundreds of frames can be consistent with at least some properties of the observed data. The search for a best overall interpretation can be effective only if many properties of the data, providing multiple sources of

constraints, are considered simultaneously.

THE PARTIAL MATCH ADMISSIBILITY CRITERION

Any proposed algorithm for partial-matching two structures A and B ought to satisfy the following criterion:

> The more similar A and B are (everything else held constant), the faster the partial match should be.

This criterion is called the partial match admissibility criterion. Its reasonableness and desirability are intuitively apparent. Yet, even in the simplest applications of partial matching, it is rarely achievable [33]. The cause is that typical partial matching algorithms evaluate properties one-at-a-time. For example, if we wish to find a document that has keys (attributes) g, h, and k, most procedures accomplish this by intersecting the inverted lists of documents associated with each of the three keys. Thus, it takes longer to find a document that matches 10 keys than to find one that matches 3, and so forth.

Avenues of approach toward realizing admissible algorithms are suggested by considering partial matching as a search problem in which each partial match corresponds to a state. The initial state is represented as a three-tuple, ((), A, F), where A is the observed data representation (or query) and F is a set of frames against which A can be compared. As before, the first component represents the abstraction or partial match thus far constructed,

the second component represents the residual of A with respect to this abstraction, and the third component represents the residuals of the frames vis-a-vis the current abstraction.

By applying typical admissibility criteria of general searches [30], it is apparent how one should move through this search space. At each decision point in the algorithm, the most promising partial solution should be extended. The most promising extension is the one providing the most complete partial match for the least expense. Here, expense is defined as the total computation required to arrive at any given state, including both the computation time spent developing the particular partial match as well as the time spent constructing collateral matches from expanded partial solutions on the same path. Thus, the best step at each point is the one which adduces the most constraint for the least cost. Constraint in this case is exactly definable as the reduction in the remaining uncertainty regarding which frames of F are involved in the best match of A.

From this viewpoint, it appears that there is only one interpretation of constraint. A transformation from one partial matching state to another is constraining to the extent to which it eliminates possible elements of F from further consideration.

Two useful concepts in this context are the diagnosticity of a test and its performance. Diagnosticity is a measure of the ability of a test to rule out possibilities. Performance is a

composite measure of the expected utility of a test, combining its diagnosticity with its expected frequency of satisfiability [8]. An optimal algorithm would apply, at each decision point, the most diagnostic test that is satisfiable. Expected cost can be minimized by applying the tests with highest performance values at each decision point. Such an approximation is important, because we know of no reasonable way to determine dynamically the most diagnostic tests. Some avenues of approach to these problems are suggested in the next section.

IMPLICATIONS FOR THE DESIGN OF KNOWLEDGE SYSTEMS

From this study of partial matching, four general implications for the design of knowledge systems are drawn. Each of these is considered in turn.

Analyses should be synthetic and dynamic. This criterion, although sounding superficially like a suggestion for analysis-by-synthesis, is diametrically opposed to that approach. In analysis-by-synthesis [19], patterns are interpreted by top-down methods: one most likely, highest-level frame is selected arbitrarily to apply and, at each point, unfilled frames are expanded downward until they can fit (interpret) the data. Because such search strategies are insensitive to properties of the data at hand, they will perform badly unless more constraint is available from the top-down structure of the frame system than from tests based on diagnostic combinations of data and

frames. To be synthetic means choosing tests to perform which, in view of the properties exhibited by the data, apply maximal constraint. Knowledge systems designed along these lines would employ a basic three-step cycle: (1) a small number of highest-performance tests are applied to the best partial solutions (initially, to the most credible data); (2) the most promising matches are extended; and (3) the new test matches are identified for evaluation by another set of highest-performance tests. Note how this paradigm embraces the WOSEQ-PPARSE methodology described earlier.

Descriptions should be rich and simple. To reduce the complexity of the search problem, descriptions should be as rich and simple as possible. This criterion implies that high-level descriptors are more desirable than low-level ones. For example, language processing systems representing knowledge in terms of lexemes are more efficient than those representing such knowledge in the form of equivalent graphs of semantic primitives [7]. One particularly interesting aspect of Merlin is its use of hierarchical descriptions permitting partial matching to be performed at the highest-level of description possible. Merlin's partial matcher descends into the depths of low-order descriptions only if matches of rich, high-level terms fail. This criterion is actually a heuristic for achieving maximally constraining tests for the least cost. Its actual effectivness depends on the exact performance of tests at high and low levels; in reasonable problem domains, however, the heuristic should be

generally valid.

Scheduling of computational resources, based on diagnosticity or performance, should be considered a primitive function in partial matching systems. Complex partial matching systems must include mechanisms to insure that the most desirable actions are executed first. Two properties of schedulers are proposed. First, desirability should primarily reflect the diagnosticity of a pending action. Second, since scheduling is a primitive operation, the costs of calculating desirabilities and sorting the pending actions should be minimized. In this context, it is interesting to note that previous studies of knowledge system scheduling [13] and conflict resolution in production systems [23, 29] have completely neglected the concept of diagnosticity.

Problem representations should integrate characteristics of the knowledge base with properties of the data to maximize the constraint provided in search. This criterion suggests that one approach to improved performance in partial matching is to develop globally organized representations whose attributes can be exploited to reduce uncertainty during partial matching. The work of Marr and Poggio [22] on stereo disparity is a good example of the use of such a globally organized problem space. Each locus of computation is influenced by all relevant cooperative loci, and these are efficiently identifiable because they are in the same

neighborhood of the problem space. The essence of such spatial organizations is an ability to reduce the number of computations involved in similarity judgments. Similar benefits were provided to the partial matcher in Merlin as a result of its hierarchical organization of knowledge.

In the future, representations should be sought which support the use of proximity measures or directionality to identify good partial matches. These could provide cheap and constraining tests for a variety of tasks. For example, semantic networks might be superimposed upon the type of metric semantic spaces which humans apparently possess [32, 34, 39]. The value of such organizations would derive from an improved capacity to detect that two objects are likely correspondents (are highly similar) just because they are close in the metric representational space. Moreover, such integrated spatial and symbolic representations could significantly improve intersection searches by favoring spread of activation in the "area" between two concepts of interest. Given the coordinates of two nodes to be connected by a best path, preference should be given to out-going links that are oriented in appropriate directions.

Other types of organization should also be sought that can facilitate computation of approximate similarity. For example, in early experiments in rule induction, Hayes-Roth and McDermott [15] showed how transformational grammar rules could

be inferred by partial-matching before-and-after examples. Their program employed no knowledge about either the structure of productions or sentences. By incorporating properties of these structures as attributes of the representations, Vere was able to reduce the computation time by two orders of magnitude [38]. The organizing properties he exploited included a three-part decomposition of each production, corresponding to the three components of the partial match of the before and after parts of each example, and a hierarchical representation of sentences. The additional constraints provided by these global attributes of problem organization greatly simplify this particular partial matching problem.

CONCLUSIONS

I have tried to show in this paper that partial matching is central to many interesting functions of knowledge systems. A few years ago, the foremost problem of knowlege system design was how knowledge should be represented. While knowledge representations are continually improving, many good frameworks have already been developed. Since pattern-directed function invocation is obviously desirable for many applications of these knowledge systems, attention has recently focused upon good methods to invoke appropriate knowledge units. Within the framework of all-or-none knowledge application, the major topics of interest concern matters of efficiency, such as developing methods for common subexpression elimination, efficient

techniques for all-or-none pattern matching, and strategies for conflict resolution. While these are surely important considerations in implementing systems for simple or well-structured tasks, the most difficult problem arising in very large and flexible knowledge systems is to determine, as quickly as possible, the most useful knowledge for the task at hand. Because many diverse elements of knowledge may be weakly contributory to an overall solution, new ways of organizing computation must be developed to prevent intractable, combinatorial searches. In the future, a major shift in attention can be anticipated toward the deceptively easily stated but fundamental question: How should partial and best matches be computed?

REFERENCES

1. Barrow, H. G., and Tenebaum, J. M. MSYS: a system for reasoning about scenes. Stanford Research Institute, Menlo Park, 1976.

2. Botrow, D., and Winograd, T. A knowledge representation language. Cognitive Science, in press.

3. Brown, J. S. Steps toward automatic theory formation. Proceedings of the Third International Joint Conference on Artificial Intelligence. Stanford, 1976.

4. Bruner, J. S., Goodnow, J. J., and Austin, G. A. A Study of Thinking. Wiley, New York, 1956.

5. Burge, J., and Hayes-Roth, F. A novel pattern learning and classification procedure applied to the learning of vowels. Proceedings of the 1976 I.E.E.E. International Conference on Acoustics, Speech, and Signal Processing. Philadelphia, 1976.

6. Evans, T. G. A program for the solution of geometric-analogy test questions. In Semantic Information Processing, Minsky, M. (Ed.). MIT Press, Cambridge, 1968.

7. Hayes-Roth, B., & Hayes-Roth, F. The prominence of lexical information in memory representations of meaning. Journal of Verbal Learning and Verbal Behavior, 1977, in press.

8. Hayes-Roth, F. Schematic classification problems and their solution. Pattern Recognition, 6, 1974, 105-114.

9. Hayes-Roth, F. Patterns of induction and associated knowledge acquisition algorithms. In Pattern Recognition and Artificial Intelligence, Chen, C.H. (Ed.). Academic Press, New York, 1976.

10. Hayes-Roth, F. Uniform representations of structured patterns and an algorithm for the induction of contingency-response rules. Information and Control, 32, 1976, in press.

11. Hayes-Roth, F., and Burge, J. Characterizing syllables as sequences of machine-generated labelled segments of connected speech: a study in symbolic pattern learning using a conjunctive feature learning and classification system. Proceedings Third International Joint Conference Pattern Recognition. Coronado, California, 1976.

12. Hayes-Roth, F., Erman, L. D., Fox, M., and Mostow, D. J. Syntactic processing in Hearsay-II. In Speech understanding

systems: summary of results of the five-year research effort. Department of Computer Science, Carnegie-Mellon University, Pittsburgh, 1976.

13. Hayes-Roth, F., & Lesser, V. R. Focus of attention in a cistributed logic speech understanding system. Proceedings of the 1976 I.E.E.E. International Conference on Acoustics, Speech and Signal Processing. Philadelphia, 1976, 416-420.

14. Hayes-Roth, F., & McDermott, J. Learning structured patterns from examples. Proceedings of the Third International Joint Conference on Pattern Recognition. Coronado, California, 1976.

15. Hayes-Roth, F., & McDermott, J. Knowledge acquisition from structural descriptions. Department of Computer Science Working Paper, Carnegie-Mellon University, Pittsburgh, 1976.

16. Hayes-Roth, F., Mostow, D. J., and Fox, M. S. Understanding speech in the Hearsay-II system. In Natural Language Communication with Computers, Bole, L. (Ed.). Springer-Verlag, Berlin, 1977.

17. Hirschman, L., Grishman, R., and Sager, N. Gramatically-based automatic word class formation. Information Processing and Management, 11, 1975, 39-57.

18. Hunt, E. B. Concept Formation: An Information Processing Problem. Wiley, New York, 1962.

19. Hunt, E. B. Artificial Intelligence. Academic Press, New York, 1975.

20. Lesser, V. R., Fennell, R. D., Erman, L. D., and Reddy, D. R. Organization of the Hearsay-II speech understanding system. I.E.E.E. Transactions on Acoustics, Speech and Signal Processing, ASSP-23, 1975, 11-33.

21. Lesser, V. R., Hayes-Roth, F., and Birnbaum, M. The word sequence hypothesizer in Hearsay-II. Proceedings of the I.E.E.E. International Conference on Acoustics, Speech, and Signal Processing, 1977.

22. Marr, D., and Poggio, T. Cooperative computation of stereo disparity. MIT AI Laboratory Memo 364, Cambridge, 1976.

23. McDermott, J. and Forgy, L. Conflict resolution strategies, Computer Science Department, Carnegie-Mellon University, Pittsburgh, 1977.

24. McDonald, D., and Hayes-Roth, F. Inferential searches of knowledge networks as an approach to extensible language understanding systems. In Pattern-Directed Inference Systems, Waterman, D.A. and Hayes-Roth, F. (Eds.). Academic Press, (in press).

25. Michalski, R. S. AQVAL/1--Computer implementation of a variable valued logic system VL1 and examples of its application to pattern recognition. Proceedings First International Joint Conference on Pattern Recognition, 1973.

26. Minsky, M. A framework for representing knowledge. In The Psychology of Computer Vision, Winston, P.H. (Ed.). McGraw-Hill, New York, 1975.

27. Mostow, D. J., and Hayes-Roth, F. A production system for speech understanding. In Pattern-Directed Inference Systems, Waterman, D.A. and Hayes-Roth, F. (Eds.). Academic Press, (in press).

28. Moore, J., and Newell, A. How can Merlin understand? In Knowledge and Cognition, Gregg, L.W. (Ed.). Erlbaum, New York, 1974.

29. Newell, A., A theoretical exploration of mechanisms for coding the stimulus. In Coding Processes in Human Memory, Melton, A.W. and Martin, E. (Eds.). Winston and Sons, Washington, D.C., 1972.

30. Nilsson, N. Problem solving methods in artificial intelligence. McGraw-Hill, New York, 1971.

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

32. Rips, L.J., Shobin, E.J., and Smith, E.E. Semantic distance and the verification of semantic relations. Journal of Verbal Learning and Verbal Behavior, 12, 1973, 1-20.

33. Rivest, R. Partial-match retrieval algorithms. SIAM Journal on Computing, 5, 1976, 19-50.

34. Shepard, R.N., Kilpatrick, D.W., and Cunningham, J.P. The internal representation of numbers. Cognitive Psychology, 7, 1975, 82-138.

35. Stoffel, J. C. A classifier design technique for discrete variable pattern recognition problems. I.E.E.E. Transactions on Computers, C-23, 1974, 428-441.

36. Tretiakoff, A. Computer-generated word classes and sentence structures. Information Processing 1974, Proceedings of IFIP

Congress 74, 1974.

37. Vere, S. A. Induction of concepts in the predicate calculus. Proceedings Fourth International Joint Conference Artificial Intelligence, 1975.

38. Vere, S.A. Inductive learning of relational productions. In Pattern-Directed Inference Systems, Waterman, D.A. and Hayes-Roth, F. (Eds.). Academic Press, (in press).

39. Voss, J. F., Bisanz, G., and LaPorte, R. Semantic memory and semantic context. Paper presented at the Annual Meeting of the Psychonomic Society, St. Louis, 1976.

40. Winston, P.H. Learning structural descriptions from examples. In The Psychology of Computer Vision, Winston, P.H. (Ed.). McGraw-Hill, New York, 1975.