May 1983

Incremental Partial Matching of Descriptions in a Knowledge Representation Network

Robert M. Chafetz
University of Pennsylvania

Recommended Citation

Robert M. Chafetz, "Incremental Partial Matching of Descriptions in a Knowledge Representation Network", . May 1983.

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-83-15.

Incremental Partial Matching of Descriptions in a Knowledge Representation Network

Abstract

This thesis describes a description matcher for the KL-ONE knowledge representation language that is incremental and can perform partial matches. The matcher is incremental in that it can be suspended at any time, producing partial results, and later restarted. The matcher is partial in that it has a formalism for measuring the degree to which two descriptions match. Both features are supported by using four different scalar metrics for the degree of match of two descriptions: a lower bound, an upper bound, a base value, and a current estimate.

Comments

University of Pennsylvania Department of Computer and Information Science Technical Report No. MS-CIS-83-15.

INCREMENTAL PARTIAL MATCHING OF DESCRIPTIONS IN A KNOWLEDGE REPRESENTATION NETWORK

Robert M. Chafetz

MS-CIS-83-15

Department of Computer and Information Science
Moore School/D2
University of Pennsylvania
Philadelphia, PA 19104

May 11,1983

INCREMENTAL PARTIAL MATCHING
OF DESCRIPTIONS IN A
KNOWLEDGE REPRESENTATION NETWORK

Robert M. Chafetz

TR-MS-CIS-1983-15

Department of Computer and Information Science
The Moore School
University of Pennsylvania
Philadelphia, Pennsylvania 19104

May 11, 1983

ABSTRACT

This thesis describes a description matcher for the KL-ONE knowledge representation language that is incremental and can perform partial matches. The matcher is incremental in that it can be suspended at any time, producing partial results, and later restarted. The matcher is partial in that it has a formalism for measuring the degree to which two descriptions match. Both features are supported by using four different scalar metrics for the degree of match of two descriptions: a lower bound, an upper bound, a base value, and a current estimate.

Index of Figures

CHAPTER 1

INTRODUCTION

This thesis describes a description matcher for the KL-ONE knowledge representation language that is incremental and can perform partial matches. The matcher is written in Franz Lisp and uses Franz Lisp translations of Interlisp KL-ONE functions (Finin, 1982). The matcher is incremental in that it can be suspended at any time, producing partial results, and later restarted. The matcher is partial in that it has a formalism for measuring the degree to which two descriptions match. Both features are supported by using four different scalar metrics for the degree of match of two descriptions: a lower bound, an upper bound, a base value, and a current estimate.

1.1 PREVIOUS MATCHERS

Matching plays a central role in description manipulation systems. Evans (Evans, 1968) and Winston (Winston, 1975) used matchers to solve analogy and learning problems. Many of the match methods used in this matcher are related to techniques implemented or proposed in MERLIN (Moore, 1973), KRL (Bobrow, 1977a,b), and FFRL (Finin, 1980) matchers, as well as a recent study of the partial match problem (Holmes, 1981). Related specifically to the KL-ONE language, Woods (Woods, 1979) proposed a Most Specific Subsumer and other algorithms to solve a variety of "situation recognition" problems and Lipkis (Lipkis,1981) has developed a KL-ONE classifier. The following sections outline some of the significant features of these matchers and classification algorithms.

1.1.1 KRL

The description matcher proposed and described in the KRL paper addressed a number of issues involved in matching intensional prototype descriptions. The act of comparison, fundamental to matching, is also a key principle of the KRL representation which "emphasized the importance of describing an entity by comparing it to another entity described in the memory" (Bobrow, 1977a, p7). The match process involves determining if a description object fits a pattern description. Each "unit" description is composed of a number of slots. The slots of the pattern and object are aligned and their fillers are matched as sub-task matching processes are set up. There is an "ability to attach specialized matching procedures to descriptions and units" (Bobrow,1977a,p25) and parallel processing of sub-tasks, considering computational resource allocation priorities is possible. A "forced match" capability is proposed, where a list of conditions to satisfy to allow a successful match would be returned. A division of knowledge into disjoint "basic" categories allows initial checking of description relations.

The KRL matcher can be suspended when resources have been depleted to return a partial match result. There is a suggestion for a "goodness measure" and reliability measure for partial matches, although the exact scheme for determing such quality evaluation terms is not delineated.

1.1.2 FFRL

The process of semantic interpretation of compound nominals as described in (Finin,1980) is match intensive. There a concept matcher "determines whether the first (concept) describes the second and, if it does, how well" (Finin,1980,p2) . Frames, instead of units, are used to represent concepts. As with the KRL matcher, slot-alignment and recursive slot filler matches are performed. Alignment is by pairing slot names, although there is a suggestion for the use of a more general structural alignment if the slots are hierarchically organized. The FFRL matcher allows invocation of specialized match procedures, as well as a most general "fms-recurse" recursive approach. There are match procedures that:

1. check for equality

2. check if the target is a sub-concept of the pattern

3. check if the match was performed previously

4. discover mismatches because of "basic" categorization

5. match slots without values

6. prevent infinite looping self-referential matches

7. match slots with one or more values

8. match intensional "requirements" of slots

9. perform recursive matches

Match scores result from match attempts and indicate if a positive match or a mismatch occurred, a score, bindings, and the match type used. The score is based on facet matching. For example, default and typical slot values that match score higher than prefered slots. Facets marked with varying degrees of salience contribute a score in accordance with the slot's prominence with respect to the concept as a whole. The total score reflects a degree of match and mismatch as a single number, since successful slot matches increment the score and unsuccessful ones decrement the score.

1.1.3 HOLMES PROJECT

A senior project by Peter Holmes (Holmes,1981) proposes a partial matcher that returns four quality scores that measure the degree of match of a target concept to a pattern concept. A uniform recursive match procedure is described in which slots are aligned and recursively matched. Multiple alignments of target roles to pattern roles may occur as a slot tree attached to a concept tree is assumed to exist. Best slot alignments are determined by factoring in match scores and subsumption distances between the pattern role and the alternative target roles.

Weights are assigned to slots, similar to the salience measures of the FFRL matcher. In this case, however, the sum of a concept's slot weights is 1. Certain slots may be marked as obligatory, meaning those roles' contributions are added only if their fillers match exactly. The four score measures used are a lower bound, base, estimate, and upper bound. These also serve as the basis for the match scores used in our partial matcher. Match-Task score and status information is maintained for each pattern concept matched and match results found from "bottomed out" returns are triggered up to the higher level Match-Tasks. The match may be suspended if processing constraints so require.

The general outline of a match process developed in Holmes' paper has served as a significant basis for the match approach further detailed in this thesis.

1.1.4 KL-ONE CLASSIFICATION ALGORITHMS

Work by Woods (Woods,1979) and Lipkis (Lipkis,1981) discusses "assimilating new concepts into the network". In the process of finding the correct place for a new description, recursive subsumption techniques , similar to those in this partial matcher, are used.

Woods describes a number of subsumption algorithms, including MSS (Most Specific Subsumer), MGS (Most General Specializer), and MSMGU (Most Specific Most General Unifier) algorithms. The MSS algorithm is intended to find the most specific generalization of the input concept; this is the place where the new concept should be attached. A concept subsumes another concept if SuperC links join them or if all role V/Rs of the pattern concept subsume corresponding roles of the target concept.

Lipkis extends this definition of subsumption to include number restriction and structural description subsumption. Both Lipkis and Woods are concerned with efficient means of searching knowledge networks to discover concepts most closely related to an input concept. The partial match problem is concerned with the degree of match between two concepts whose parts and relationships have already been made explicit through such classification schemes.

1.2 OVERVIEW

Our matcher returns partial scores, from 0 to 1, as measures of the degree to which some "target description" may be viewed as a given "pattern description". A pattern concept matches a target concept as its roles are put in functional correspondence with the target concept´s roles and its role fillers are recursively matched against the target role fillers. The roles of a pattern can be assigned weights which indicate the importance or salience of each role with respect to the concept as a whole. When two roles are matched, the amount contributed to the score of the overall match is a function of the pattern role´s salience, a determination of functional alignment, and the degree of match of the role fillers.

The degree to which two descriptions match or can match is represented by four scores. Two of these, a lower and upper bound, are related to the "evidential propositional calculus" (Garvey,1981) based on Shafer´s theory of evidence (Shafer,1976) .The lower bound is evidence for a match, and the upper bound is 1 - evidence against a match.

Scores of a partially completed match process are current best estimate scores since all score information found at termination objects at the bottom of the network are triggered to the top of the network. Match-Tasks, which are types of meta-descriptions represented in the KL-ONE formalism, contain match score and status information. A Match-Task is created and maintained for each pattern concept reached in the match process. In this way the match process may be suspended and restarted from its previous state of comparison.

CHAPTER 2

KL-ONE REPRESENTATION

2.1 INTRODUCTION

The KL-ONE knowledge representation language, a semantic network formalism built on the use of epistemological primitives (Brachman,1979a), is used to describe intensional data constructs. Generic concepts, which represent general intensional descriptions, form the skeletal structure of the network. Conventions for concept structuring, role inheritance, inter-role relationships, and concept individuation are part of the rules for the language. Thus a system for concept description exists independent of a particular domain's semantic interpretation.

In order for the constructed networks to be interpretable with respect to a possible world, it is the responsibility of the Knowledge Engineer to make correspondences between network concepts and possible objects and network roles and possible object attributes.

2.2 CONCEPTS

2.2.1 INTRODUCTION

Concepts are intensional in that they describe a class of potential objects. They are not associated with actual world objects and events, which are represented by other means (Woods,1975). Individual concepts are also intensional, but represent unique concepts that individuate Generic Concepts. Generic Concepts may be satisfied by a large number of Individual Concepts. These Individual Concepts can be considered as unique variants of a generic prototype. They remain intensional since there is not necessarily a real-world extensional object associated with each of them and two individual concepts may refer to the same extensional object .

2.2.2 GENERIC CONCEPTS

Generic Concepts may be related to each other in a graph network where nodes are the generic concepts and the links are SuperC arcs. A sub-generic concept specializes a super-generic concept. The sub-concept "is a" restricted case of the super-concept. The sub-concept may be linked to more than one super-concept, thus taking on a conjunction of those super-concepts' properties.

Figure (1) exemplifies a generic concept hierarchy, where a generic concept may have more than one ancestor and individual concepts appear as leaf nodes attached to generic concepts. The key of figure (8) , which follows Brachman's graphic sign conventions (Brachman,1978c), serves as a guide for the KL-ONE network diagrams in this thesis. Figures (7) and (22) serve as reference examples from which may be observed the KL-ONE object organization and relationships as the formalism is described in the following sections.

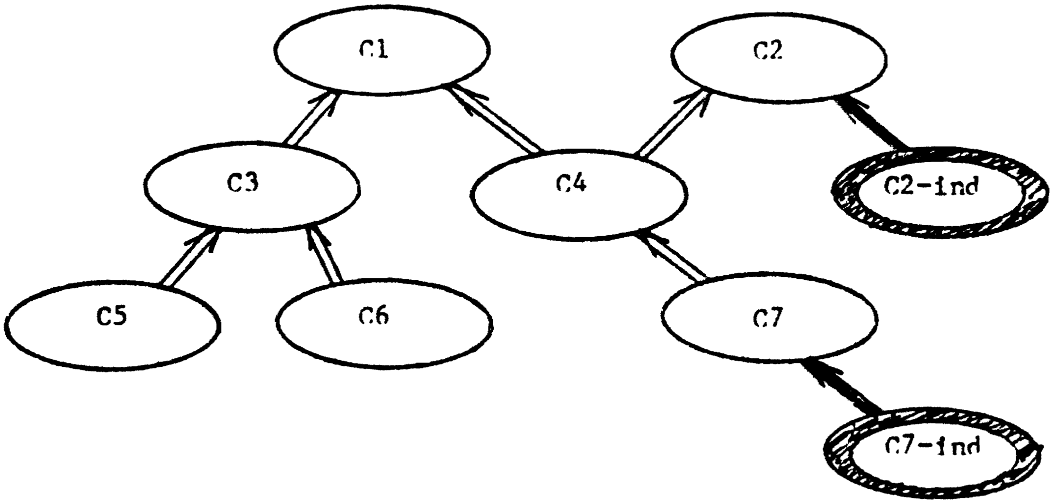


Figure (1) Generic Concept Hierarchy with Individual Concept Leaf Nodes

A recursive description may not be constructed by using SuperC arcs to make a concept both a sub-concept and super-concept of another concept, as shown in figure (2).

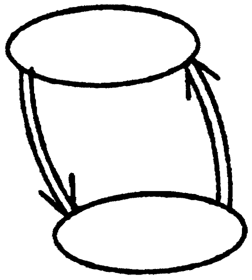


Figure (2) Illegal Recursive Description

2.2.3 INDIVIDUAL CONCEPTS

An individual concept is a specific case of its generic concept, where particular parts of the generic concept are filled by other individual concepts. They are necessarily leaf nodes in the graph, since they may not be further modified. As discussed in the next sections, the full meaning of concept specialization and individuation is seen with respect to the roles and structural descriptions of the concepts.

2.3 ROLES

2.3.1 INTRODUCTION

Roles serve to characterize the parts or attributes of a concept. There are generic roles for generic concepts and instance roles for individual concepts. Roles exist only as parts of their owning concepts; they may not be defined outside of the context of a concept.

2.3.2 GENERIC ROLES

Generic roles are intensional objects that describe the class of concepts that can serve in that function, the number of those concepts, and the need for those fillers to properly describe the owning concept.

2.3.3 ROLE FACETS

A Generic Role has three role facets:

1. VALUE RESTRICTION (V/R)

2. NUMBER

3. MODALITY *

Figure (3) shows the components of a generic role.

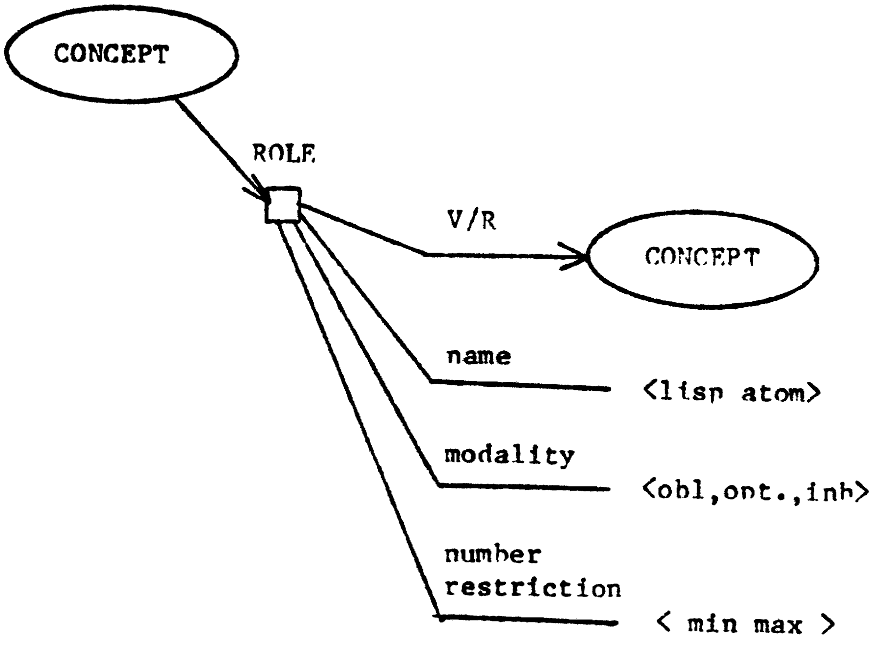


Figure (3) Role Facets

2.3.3.1 Value Restriction -

A generic concept´s value restriction is a generic concept. This is the type of filler the role is expected to have.

* The use of a role modality has been dropped from the current version of KL-ONE.

2.3.3.2 Number -

A generic concept´s Number is a pair of non-negative integers that designate the minimum and maximum bounds of the number of individual concept value restrictions a satisfying instance role must have.

2.3.3.3 Modality -

A generic role´s modality specifies whether a valid individuator must or may have an instance of that role. Obligatory generic roles must be satisfied by an instance role whose value is an individual concept that individuates the generic value restriction´s generic concept. This means that the generic role´s Number´s minimum bound must be greater than 0. An optional generic role allows the possibility of an instance role existing, but it need not be present to form a valid individuator. This implies that the optional generic role´s Number may be thought of as a set containing 0 and the values ,inclusive, from the minimum to the maximum.

2.3.3.4 Role Names -

Each role has one or more role names. The names of roles serve as identifiers for the user. Different roles may have the same name. For example, a sub-concept´s role may serve the same function as a super-concept´s role and therefore be most naturally referred to with the same name. A role may have multiple names if it is inherited from a number of super roles.

2.3.4 INSTANCE ROLES

An instance role is a bottom-level KL-ONE object that is linked to a generic role that it satisfies and has a value that is a bottom-level individual concept. An instance role may not be further specialized, and is attached to an individual concept.

2.3.5 ROLE INHERITANCE

2.3.5.1 Introduction -

Structured inheritance is a key mechanism of KL-ONE . It is the means by which parts of a more general descriptional network become the descriptional framework of a more specialized network. There are three basic ways that roles may be connected to their ancestor roles: modification, differentiation, and satisfaction. When a sub-generic concept specializes a super-generic concept, the sub-roles may be in a modification or differentiation relation with the super-roles. When an individual concept Individuates a generic concept, the instance roles of the individual concept Satisfy the generic roles. Any generic roles of a super-concept that are not explicitly modified or differentiated at a sub-concept will be inherited as exact "virtual copies" (Fahlman,1979).

2.3.5.2 Modification -

When a sub-role is in a modification relation to a super-role , some of its facets are further restrictions of the super-role´s facets. The sub-role´s value restriction may be a generic concept that is a subset/specialization of the super-role´s value restriction. For example, from figure (7), File-Copied-To-New-Directory´s Location role modifies Object-With-Dir-Location´s Location role and Different-Directory is a restriction, or subset of the Directory concept. The sub-role´s Number may cover a range that is a subset of the super-role´s Number.

2.3.5.3 Differentiation -

A sub-role in a differentiation relation with a super-role may also have local restrictions that override the super-role´s facets, as in the modification relation. Differentiation is used primarily to form a set of sub-roles of a super-role. The differentiated sub-roles may be attached to super-roles and considered as owned by the super-concept.

2.3.5.4 Satisfaction -

This relation exists between an instance role of a sub-concept and a generic role of a super-concept. It serves as a place from which to refer to individual concept fillers.

2.3.5.5 Multiple Role Inheritance -

When a sub-concept has multiple ancestors, two or more functional roles of the ancestors may be described as a single functional role of the sub-concept. The default interpretation of the sub-role´s value restriction is a conjunction of the super-roles´ value restrictions.

2.4 STRUCTURAL DESCRIPTIONS

The formalism, as presently described, has roles which are explicit objects that refer to concepts. Roles are meant to be a functional part of a concept or a relation in which a concept is involved. There is some structure in the way that roles are formed since sub roles are connected to super roles. However, the nature of the relationship or function the roles represent is not clear; in a sense it is left up to the knowledge engineer or user to attach meaning to the role by using the role name's evocational sense. KL-ONE addresses this issue by allowing the use of structural descriptions to make explicit the functional relationships in which roles are involved. A structural description may be thought of as a way to overlay the roles of a separate concept's roles over specified roles of the generic concept being described. The formal objects used to construct a structural description are an SD object, a paraindividual concept, and coreference roles of the parindividual. The SD is analogous to a role and the paraindividual is analogous to a value restriction. The paraindividual's coreference roles have links which show how the paraindividuated concept's relational roles correspond with the owning concept's roles. The interpretation of a structural description is that any individual concept of the owning concept must have individual concept role fillers that satisfy the specified structural constraints of the owning concept.

The paraindividual is parameterized in the sense that it is applicable only to its owning concept. It is a special type of individual concept in that it is specifying relations that must hold for particular individual role fillers. A special list structure, called a role chain, is required to make explicit the functional connection of the roles being constrained to the other roles of the owning concept.

The structural description shown in figure (4) places a constraint on the file body roles to require that any individual fillers of the two roles be the same object.

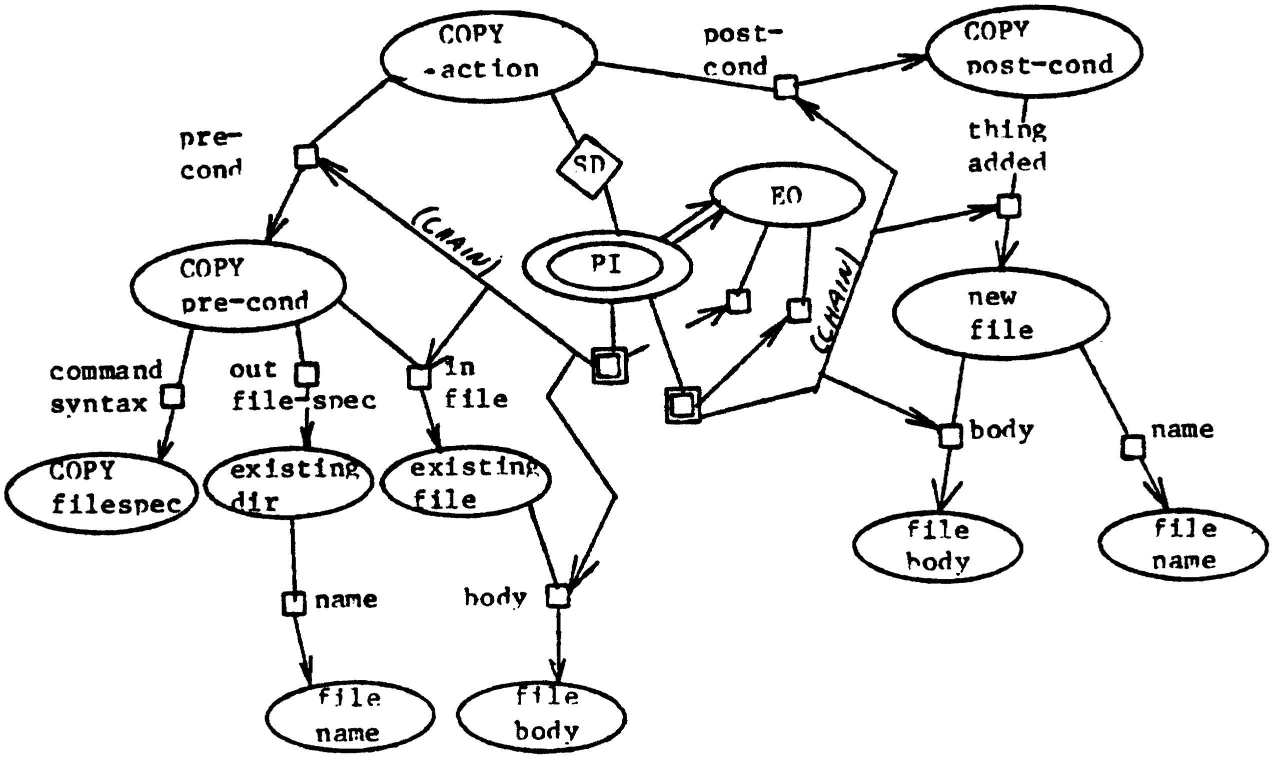


Figure (4) KL-ONE Representation of Copy Command,
Using a Structural Description

CHAPTER 3

NETWORKS TO BE MATCHED

Assumptions are made about what type of networks will be matched, and what additional datum information is associated with the concepts and roles in the network. The matcher presently does not handle structural desriptions.

3.1 VALIDATED CONCEPTS

To some extent KL-ONE enforces rules for building knowledge networks by rejecting KL-ONE functions calls if the arguments are not of the proper type. The following relations between concepts and roles are enforced automatically when the KL-ONE functions are called.

- Individual concepts may only have individual roles attached. Those individual roles may have arcs only to individual concepts.
- Generic concepts may have generic roles attached.
- Individual concepts may not be further individuated.

The following relations between concepts and roles should be enforced by the knowledge engineer to form valid KL-ONE structures:

- A sub-concept´s role´s V/R is a specialization of the super-concept´s role´s V/R when the sub-role modifies or differentiates the super role.
- A sub-concept´s role´s Number is a sub-range of the super-concept´s role´s Number when the sub-role modifies or differentiates the super role.
- A sub-concept´s role´s modality is more constrained than its super-concept´s role´s modality when the sub-role modifies or differentiates the super role.
- A subconcept´s roles only modify or differentiate roles of ancestor concepts.

3.2 CLASSIFIED CONCEPTS

Lipkis´ work, (Lipkis,1981),as mentioned earlier, describes a procedure for automatically placing new concepts in their best classified position in an existing KL-ONE network. For the partial matcher it is assumed that the pattern knowledge is best classified, either manually by the Knowledge Engineer, or with an automatic system similar to Lipkis´.

In a classified network, in the sense that Lipkis uses, a generic target role aligned by subsumption to a generic pattern role need not have its V/R recursively matched, since a sub-generic concept may not have any roles cancelled that originated from its super-generic concepts. However, there are three main cases where a recursive match of role V/Rs may be required to determine the proper match score. They are:

(1) - non-validated network
- network with UNKNOWN concept
-network with concepts not yet classified
(2) - the target role is an instance role
(3) - a subsumption distance is of interest

From this point on, any matching task or network in which any of these conditions holds true will be referred to as a recursive problem. Any problem in which any of these conditions is not true will be referred to as a non-recursive problem.

Non-Validated Network

When new information is being acquired or learned, its description evolves from a less complete and consistent state to a more complete and consistent one. At the bottom of the network inconsistent and/or incomplete descriptions may exist. For example, Woods states (Woods,1978,p39) "At some point sufficiently low in the lattice, one can begin to form inconsistent descriptions by the conjunction of incompatible concepts, the imposition of impossible restrictions, etc..."

Unknown Concepts

Incomplete information may be simulated in KL-ONE by using a distinguished concept, the UNKNOWN concept, which is a THING, and has no further specializers. The UNKNOWN concept must be specifically assigned, or else the V/R defaults to be the top-level THING concept. An example of its use is the representation of the RENAME command of figure (7), where RENAME´s syntax is UNKNOWN. Perhaps a novice user has forgotten the exact syntax needed.

Concepts Not Yet Classified

If a subconcept has a role whose V/R is a concept that does not specialize the superconcept´s role´s V/R, then the subconcept has not been fully classified. Figure (5) illustrates this case. Since a SuperC arc does not connect Sub V/R and Super V/R , a match of these roles would fail.

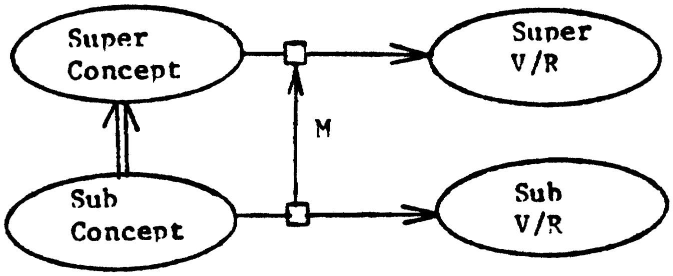


Figure (5) Non-Classified Net

Instance Target Role

The intensional nature of KL-ONE descriptions allows for the creation of individual concepts that do not have roles corresponding to optional generic roles of the individuated generic concept. For example, the target role's value, of figure (6), does not have a role corresponding to R2. When Rp's V/R sets up a sub match task, R2 will fail its match.

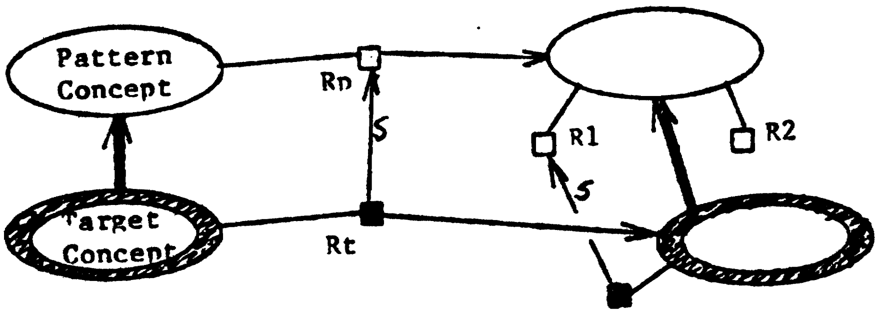


Figure (6) Network with Instance Target Role

Distance Measure Used

If a conceptual distance between a subsumer and subsumed concept can be measured it may depend on recursive measurements. At the least it will involve examining weights or datum attached to KL-ONE objects between the target and pattern roles. Section 8.2.1.1 presents a possible distance measure.

The networks to be matched for the presented matcher may be non-validated. Presently, a distance measure is not calculated.

3.3 WEIGHTING

Roles of pattern concepts are weighted to indicate the relative importance of the concept attributes with respect to the owning concept. For example, in figure (7), the creation of File-Copied-To-New-Directory is considered to contribute 60 percent of the meaning of the COPY command. It is assumed that the sum of the weights of the roles of any pattern concept will not exceed 1. If the sum of the weights is less than 1 it may be interpreted to mean that the knowledge engineer is only able to partially describe that concept.

3.4 MINIMUM ROLE SCORE

In order to allow more natural partial match specifications, a minimum score requirement for a role's value restriction is used. In figure (7), the RENAME command's Create role must have a V/R match of at least 30 percent for the Renamed-File concept in order to be partially successful. The minimum score's range is the same as a concept score range, from 0 to 1. A minimum score requirement of 0 means that any partial match is acceptable to describe the role's value restriction. If the value restriction match does not achieve the minimum score, the role match fails completely. A minimum score of 1 means that there must be an exact match or else the match attempt fails.

3.5 MODALITY

The matcher assumes that modalities of Obligatory or Optional are assigned to roles of concepts. An instance role exists intensionally and is treated similar to an obligatory role, as explained in section 5.4 .

3.6 EXAMPLE DESCRIPTION NETWORK

Figure (7) shows a KL-ONE representation of a rename and copy command. Figure (8) is a key to the KL-ONE objects and relations used in Figure (7) (Brachman,1978c). More detailed representations of commands are possible; this figure serves to illustrate the types of data and correspondences dealt with by the matcher. The RENAME and COPY concepts have roles that match in three ways:

1. EQUAL roles are inherited intact e.g. File´s Body role is inherited by Renamed-File

2. SUBSUMED roles e.g. Copy´s File´s Name subsumes Rename´s File´s Name

3. GENERALIZED roles e.g. e.g a generalization of COPY´s Word subsumes RENAME´s Word

Interpretations of the rename and copy commands are:
COPY- A command with syntax and a COPY command word. It creates a file with a file name, a file body, and a new directory location.

RENAME- A command with a RENAME command word, unknown syntax, and given first in order. It creates a file with a new file name, a file body and a directory location.

Weights and modalities have been added to the roles by the knowledge engineer to tailor the descriptions as desired. For example, the RENAME Command must have a command word RENAME, and it receives a maximum score of .3 when that role matches.

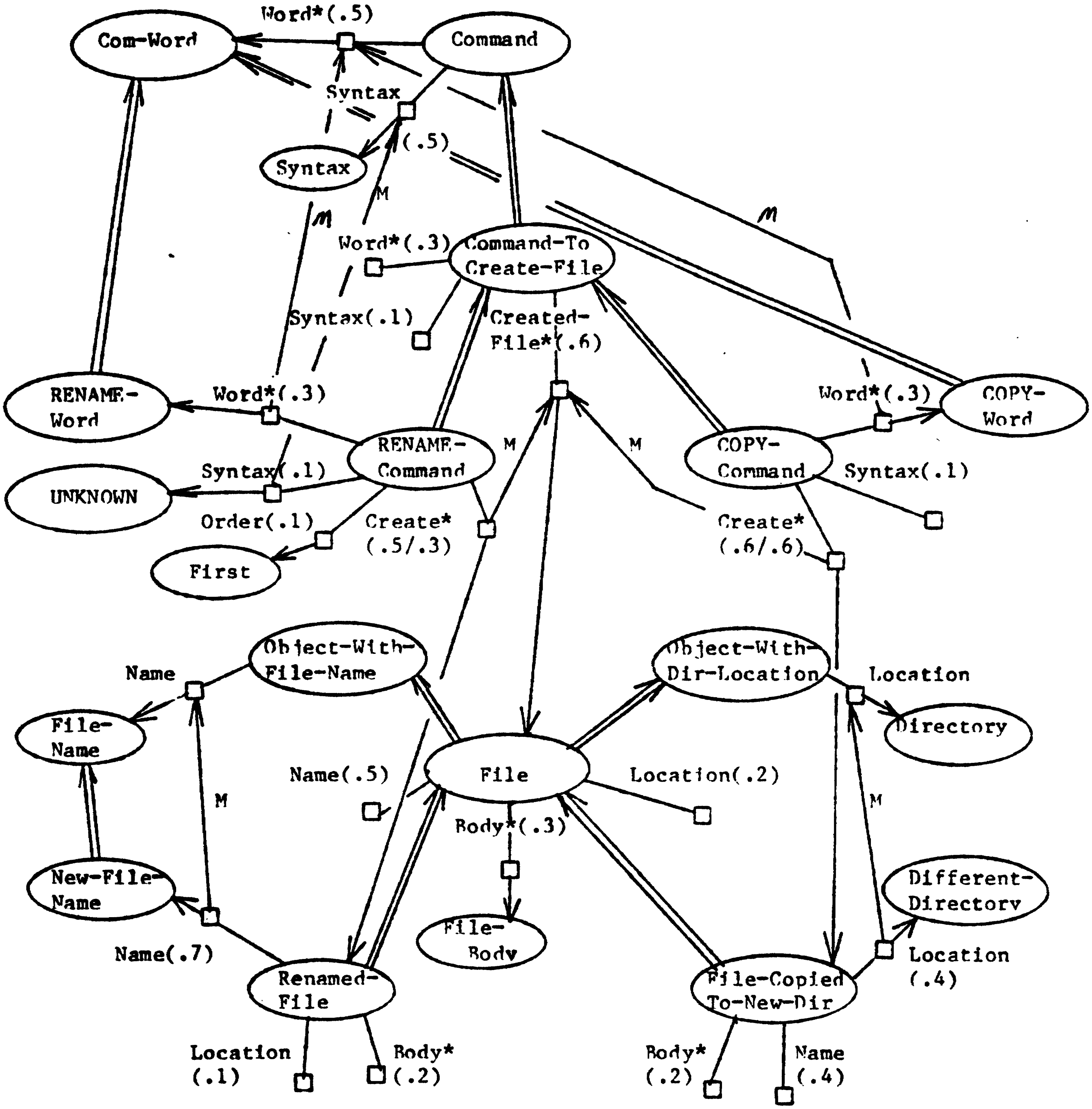
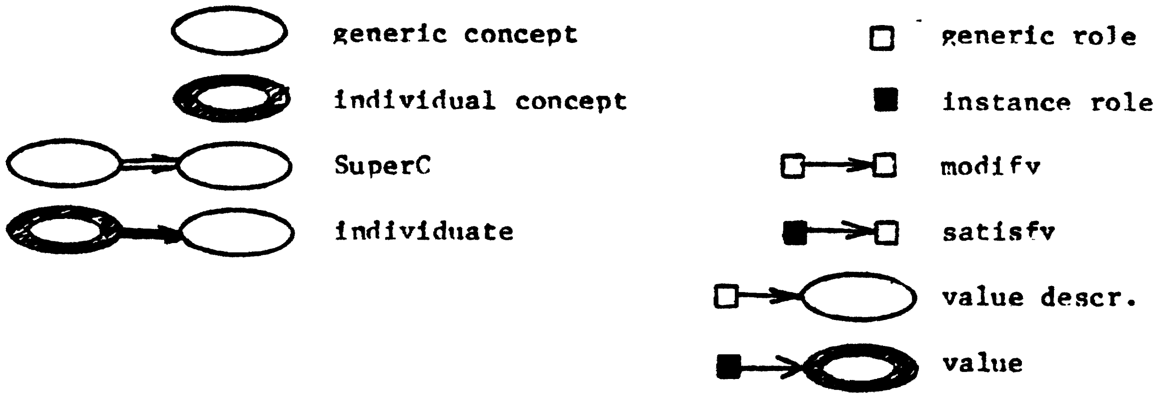


Figure (7) File command Descriptions



* == obligatory role
(<#>) == weight
(<#>/<#>) == (weight/minimum role score)

Figure (8) KL-ONE Key

CHAPTER 4

MATCH PROCESS

4.1 INTRODUCTION

The problem of determining the degree to which one conceptual description may be viewed as another conceptual description is an important element of many cognitive processes. It is a fundamental means by which new information is assimilated into frameworks of previous knowledge. Bobrow and Winograd make the following statement: "Reasoning is dominated by a process of recognition in which new objects and events are compared to stored sets of expected prototypes, and in which specialized reasoning strategies are keyed to these prototypes" (Bobrow,1977a,p5). It is also a means of comparing known descriptions in order to make explicit relationships implicit in the declarative structure.

The matching process involves traversing the nodes and arcs of the knowledge network representing a pattern concept and determining to what degree a target concept´s nodes and arcs may be put in correspondence with the pattern. The process proceeds with two-stage cyclic recursive calls for role alignment and concept matching.

A simple "succeed" or "fail" match result does not provide sufficient information to compare and contrast descriptions in a natural manner. Partial score results give a better indication of the degree and nature of the match. Partial scores, including a best estimate, lower bound, and upper bound, are updated as corresponding parts of the pattern and target concept are compared.

The matcher handles description information of the following kind:

1)<u>CLASSIFICATION KNOWLEDGE</u>:
- equality, subsumption, or generalization relations between concepts and roles of concepts
- generic or individual concept and role types

2)<u>MODALITY STATE</u>:
- obligatory, optional, existing

3)<u>PARTIAL DESCRIPTION</u>:
- minimum score to describe a concept
- role weights

The matcher is incremental and may be restarted after a match in process is suspended. As soon as a match score is determined at the bottom level of the network, scores of concepts that pointed to that concept are updated. In this way the scores at the original root pattern concept reflect the latest best estimate of the match progress. Match status datum is attached to each pattern concept before and during the match process. The match process can be restarted by examining the status of the pattern concepts until a pattern concept is found to be in an original match state.

4.2 PREPROCESSING/INITIALIZATION

A ranked role list is formed by ordering the roles with respect to their modality and weight. For example, Copy-Command´s roles ,from figure (7), would form a ranked list:

ROLES: Create*(.6), Com-Word*(.3), Syntax(.1) .

Pointers and labels are maintained to identify the target concept and how the concept was reached. The match process proceeds in a depth-first manner, matching role value restrictions in the order determined by the ranked role list.

4.3 ROLE ALIGNMENT

Role alignment is the process by which functional correspondences are made between the pattern and target roles. The explicit modification and differentiation links that exist between sub-roles and super-roles are used. Two main issues of role alignment are the types of alignment and the determination of a best alignment when multiple alignments of target roles to a pattern role exist.

4.3.1 TYPES OF ALIGNMENT

4.3.1.1 Equality -

The roles of the super-concept are inherited intact by the sub-concept if there are no explicit links from the sub-concept´s roles to the super-roles. This could be the case when the target concept is a specialization of the pattern concept. In figure (7), the Renamed-File inherits File´s Body role intact. Such roles will be shown without their V/Rs, although they are assumed to be virtually present and equal. When both the pattern and target concept are specializations of a common ancestor concept, such as the Renamed-File and File-Copied-To-A-New-Directory, each concept may have equal roles. The Body Role for these concepts is such a case.

4.3.1.2 Subsumption -

A pattern role is considered to subsume a target role if the pattern role can be reached from the target role by traversing one or more satisfaction(only one would be in the path) differentiation, or modification links. An interpretation is that a subsumed role is playing a similar role as the subsumer role, but at a lower level of abstraction. In Figure (9), for example, R1 subsumes R2,R3,and R4, R2 subsumes R3 and R4 and R3 subsumes R4.

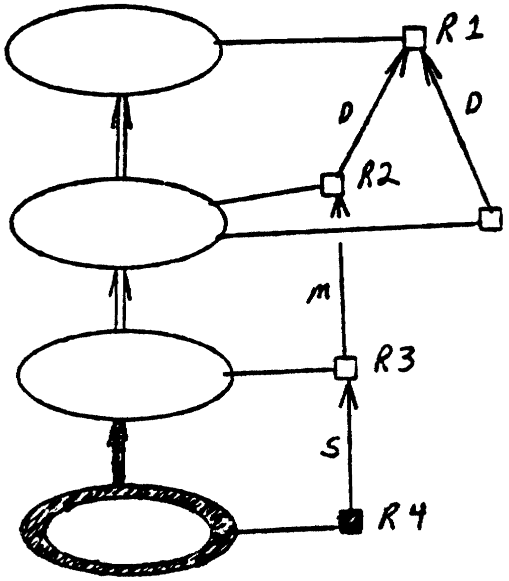


Figure (9) Role Subsumption Relations

4.3.1.3 Generalization -

If the pattern concept does not subsume the target concept, but they do have a common ancestor concept, a generalization type alignment may exist between their roles. An interpretation of this alignment is that if we generalize the pattern's function to be the ancestor function, then the generalized role subsumes the target role. For example, in figure (7), if the Word role of the COPY command is generalized to the Word role of the Command concept, it subsumes RENAME's Word role.

4.3.2 UNALIGNED ROLES

After the role alignment process, there may be pattern roles left that could not be aligned with any target roles. This results in a complete match failure of those roles. This could occur if the pattern concept owns roles that the target concept does not own. In figure (7), the RENAME command has an Order role locally attached. The COPY command does not have such a role; if it is a target for the RENAME concept, the Order role will be unaligned and fail its match. If a target role is unaligned, it is ignored. If COPY is the pattern then the Order role does not effect the match scores.

4.3.3 BEST ALIGNMENT

The alignment of target roles to pattern roles may have resulted in the alignment of more than one target role to a single pattern role. A best alignment is defined to be the one that results in the highest match score between a pattern role and one of the alternative target roles.

4.3.3.1 Subsumption -

If the target concept is subsumed by the pattern concept,multiple target roles may differentiate or modify a pattern role. For example, in figure (10), if Super is considered as the pattern concept, then Rt1, Rt2, and Rt3 all align with Rsuper. If the network is "non-recursive" , then any target roles match exactly and there is no need to find a best match. If the network is "recursive", then it is necessary to determine each pattern/target role match score and choose the highest score.

4.3.3.2 Generalization -

Figure (10) also shows a case where multiple target roles align through generalization with a single pattern role. In that case it is necessary to fully traverse the value restrictions, even if the network is non-recursive, in order to find the highest match score. This is because the V/Rs of the pattern role may have inherited roles that may not have been inherited by any or all of the target role V/Rs. For example, the pattern concept V/Rp inherits Rp´ from the ancestor concept A. Rt1 is the only target role whose V/R also inherits that role and is therefore the only role that will contribute towards the Rp´ match score. For this case, only through recursive traversals of each pattern and target role V/R networks can the highest scoring, best aligned role be found.

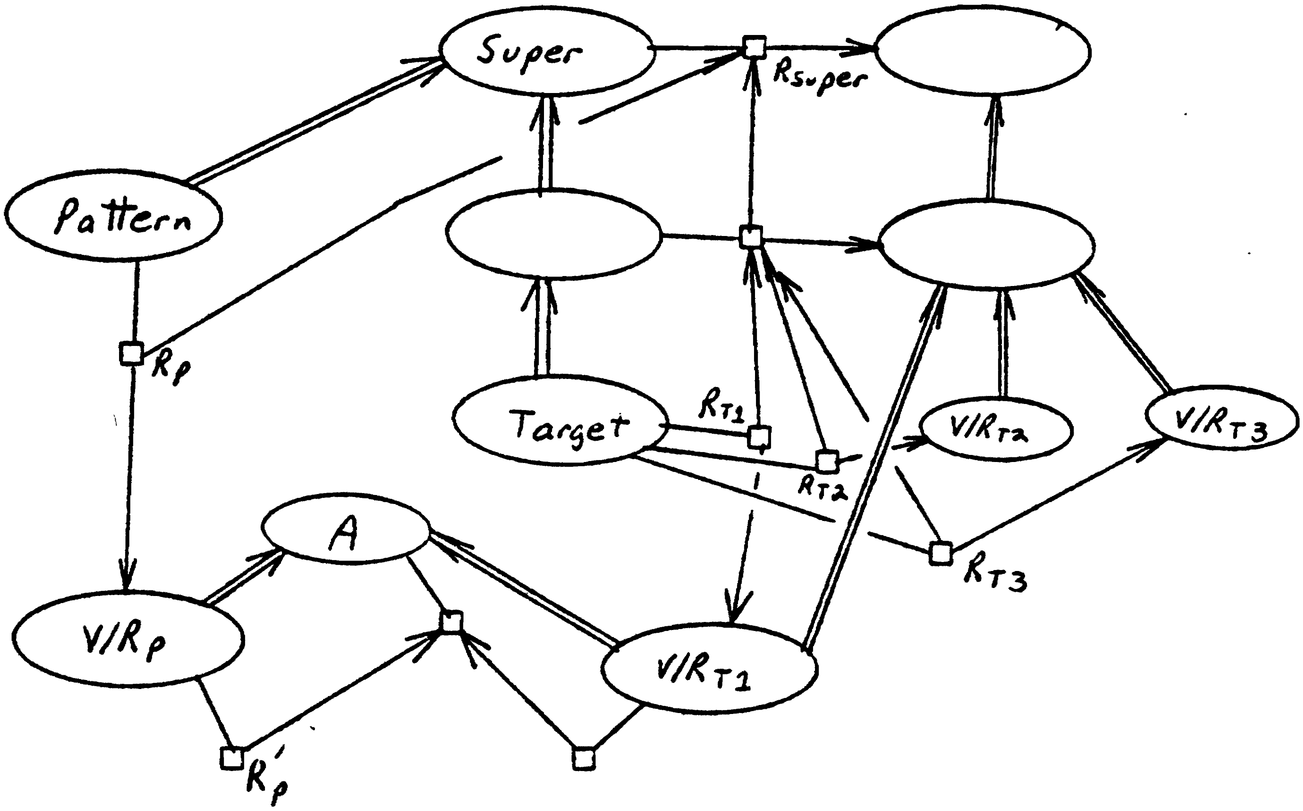


Figure (10) Multiple Role Alignments

4.3.3.3 Multiple Generalization Alignment -

Because concepts may have more than one ancestor and a role may modify or differentiate more than one super-role, a pattern role could align with a target role through multiple ancestor roles. For example, in figure (11), Rp aligns with Rt by generalization through both R1 and R2. For the present matcher, the multiple generalization alignments simply serve as alternative paths to allow matching between the same pattern and target role. The first generalization path in the alignment table is chosen ; it allows alignment of the pattern and target role as well as any other path in the sense considered here. The use of structural description information for choosing a best alignment is considered in section 8.2.2.5 .

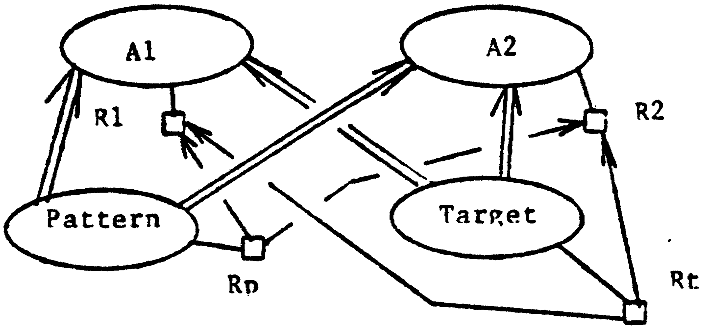


Figure (11) Multiple Generalization Alignment

4.4 ROLE FACET MATCHING

The three facets of a role all contribute towards the role´s and owning concept´s intensionality. V/R and modality relations are considered for the matcher. Number matching is considered in section 8.2.1.3.

4.4.1 VALUE RESTRICTION

Value restriction matching is performed by recursive calls to the matcher, matching the target V/R to the pattern V/R.

4.4.2 MODALITY

The descriptions being matched have attributes that must, may, or do exist. These possibilities correspond to generic obligatory, generic optional, and instance roles. When matching modality typed intensional objects, it is important to clarify the meaning of a match when the pattern and target have various modality combinations. The modality-based scoring process, to be described in section 5.3, reflects this meaning.

The type of question that arises is exemplified by asking what it means to say that a target arch that must have a lintel can be viewed as a pattern arch that may have a lintel. Here ambiguity arises since some individual pattern arches may lack a lintel and fail to match the target, while other individual pattern arches will have a lintel and will succeed with that role´s match. To circumvent this ambiguity, the interpretation for optional generic roles will be that they refer to generic terms, and not possible individuators. In this way an optional role that is present can be interpreted to mean that it is possible to describe the owning concept as having that role filled. It will be shown in section 5.3 that the use of an additional lower bound score could provide information about possible match failure of a concept´s optionally present role.

Instance roles are matched in a manner similar to obligatory roles. This is because a role that does exist, whether satisfying an obligatory generic or optional generic role, is required for the description of that unique individual.

4.5 ROLE-LESS CONCEPT MATCH

At some point in the traversal of the pattern and target concepts, concepts that do not have any roles will be reached. These are primitive descriptional units, representing the smallest "grain size" concepts in the network. If either the pattern or target concepts do not have roles, then a bottomed-out score result can be determined. There are three separate cases where the pattern or target concept may not have roles, as illustrated in figure (12) and listed in the following table.

	PATTERN	TARGET
1)	no roles	roles
2)	roles	no roles
3)	no roles	no roles

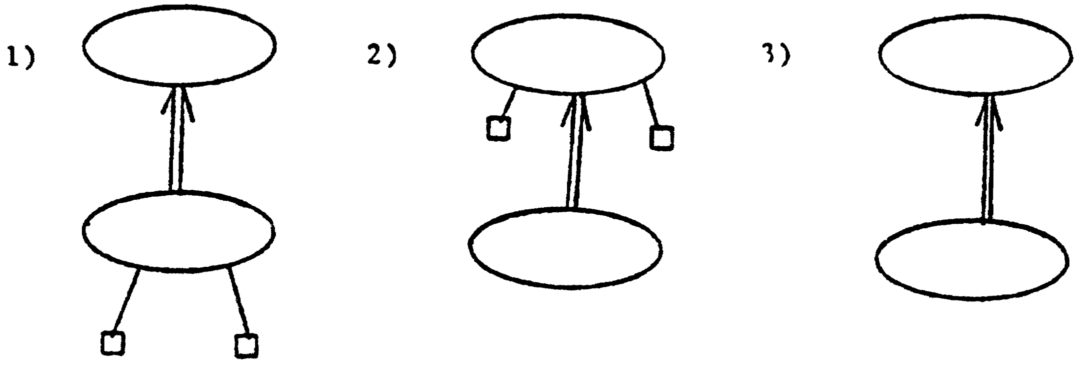


Figure (12) Concepts Without Roles

Case 1 results in success iff the pattern concept is the top level THING concept. Case 2 is assumed to always fail. Case 3 will be considered in detail in the following sections.

4.5.1 EQUALITY

The simplest type of concept match occurs between two identical generic concepts or two identical individual concepts. In the KL-ONE formalism, concepts can be identical to each other only if they are exactly the same object.

4.5.2 SUBSUMPTION

A pattern concept subsumes a target concept if the target concept is linked to the pattern concept through one or more SuperC specialization links. A generic concept is also considered to subsume an individual concept. The subsumed generic concept can be seen as a kind of super-concept and the subsumed individual concept is a unique representation of the generic concept.

If the pattern concept does not explicitly subsume the target concept through SuperC arcs, but the two concepts have a common ancestor, there may still be a partial subsumption relation between the pattern and target concept. The concepts could belong to overlapping concepts if they are not mutually exclusive concepts. The implemented matcher does not handle this case, but does establish a generalization relation between the pattern and target concept.

4.5.3 GENERALIZATION

A generalization type match can be interpreted to mean that a generalization of the pattern concept subsumes the target concept. One case is when a sub-generic pattern concept has a super-generic ancestor concept that subsumes the sub-generic target concept. Another case is when the pattern concept is an individual concept. The individuated generic concept or an ancestor of that generic concept may subsume the target concept.

4.6 PREVIOUS MATCH INFORMATION

Match-Task score and status information remains attached to the pattern concepts after the match process has been partially or totally completed. This approach is useful for avoiding match calculations that were previously performed.

CHAPTER 5

MATCH SCORING

5.1 INTRODUCTION

Match scores are recorded as part of Match-Task metadescriptions attached to all reachable pattern concepts. The score triggering process and the incremental match restart process access this stored match information. When a pattern and target concept are put in correspondence for a match, the pattern concept's Match-Task is checked to determine if the match was previously performed. If not, then the score and status information are initialized.

Match scoring involves two steps, a termination match at a sub-level, and multiple match score triggerings to super-levels. Exact match scores of 1 or 0 are determined when the match process reaches termination objects. A termination object is a KL-ONE pattern or target object that has no sub-parts, an object whose sub-parts are not fully specified, or equal objects. Such cases terminate the recursive match process and return a match score for the pattern/target pair.

If information is missing or unknown, it is simulated by using an UNKNOWN concept V/R. If either the pattern or target V/R is UNKNOWN, then no change is made to any scores. If an UNKNOWN concept is the V/R of an obligatory role, then the lower bound of the owning concept's lower bound is set and always remains at 0.

5.2 MATCH SCORES

The four concept match scores are maintained such that they are always in the following relationship:

lower bound <= base <= estimate <= upper bound .

5.2.1 ESTIMATE

A current best estimate of the concept´s score, based on partial score results of the concept´s roles, is maintained. The estimate, or score , is initially 0 and may be increased to a maximum score of 1.

5.2.2 LOWER BOUND

This is the minimum score possible for a concept, considering that partial role scores may be returned and that all roles may not have been processed yet. The lower bound is initially 0 and remains there as long as as obligatory roles remain unmatched, since the failure to match any one of the concept´s obligatory roles will cause the entire concept to fail to match. When all obligatory roles are successfully matched, the lower bound is incremented to be equal to the concept score. When the last obligatory role is being processed, all obligatory roles of that role´s V/R must have been successsfully matched, as well, in order to increase the lower bound. A role´s score must exceed its minimum role score for the lower bound to be incremented.

5.2.3 UPPER BOUND

This is the maximum score possible for a concept. The upper bound is initially 1 and is decremented whenever roles or reachable role components fail a match attempt. The upper bound falls to 0 when an obligatory role fails. A concept´s upper bound is used as a comparison threshold for determining if a role that points to that concept through a V/R link should fail its match. Whenever a role´s V/R concept´s upper bound is less than the role´s minimum score, the role should fail its match.

5.2.4 BASE

The base score is related to the lower bound but differs in that the base score is incremented whenever a role score exceeds the minimum role score and the role´s V/R has completed its obligatory role matching. The base score may increase before the lower bound does because it may be incremented even if all obligatory roles of the owning concept are not completed.

5.2.5 ROLE SCORE

Each role of a concept has its own score. The role score is the score of the role´s value restriction´s concept. The role score is initialized to 0 and is incremented as the value restriction concept score is incremented. The concept score is equal to the sum of the product of
(role score) * (role weight) for all roles processed.

5.3 MODALITY PAIRINGS AND SCORE EFFECTS

Match successes or failures of a pattern concept´s role effect the overall concept score differently depending on the modality and type of the pattern and target roles. Figure (13) summarizes these effects. The table shows that for a particular pair of modality-typed roles match successes or failures result in different effects on the overall concept scores. The four match scores implemented are lb, base, estimate, and ub. The code <S or F><#> means that a successful match(S) or failed match(F) results in score updating as indicated by the keyed number.

	pattern	target	lb´	lb	base	est.	ub
1	obl	obl	S1	S1	F1,S2	F1,S3	F1
2	obl	opt	S4	S1	F1,S2	F1,S3	F1
3	opt	obl	S4	S2	F2,S2	F2,S3	F2
4	opt	opt	(same as 3)				
5	obl	inst	(same as 1)				
6	opt	inst	(same as 3)				
7	inst	obl	(same as 1)				
8	inst	opt	(same as 2)				
9	inst	inst	(same as 1)				

SUCCESS:

S1 increase if role score > min role score ,
all obligatory role of the owning concept have been matched,
and all obligatory roles of the V/R have been matched

S2 increase if role score > min role score
and all obligatory roles of the V/R have been matched

S3 increase score

S4 score remains same

FAILURE:

F1 set score to 0 and fail concept

F2 reduce score

Figure (13) Modality Score Effects

In addition to scorings handled by the implemented matcher, the table shows a second lower bound score . This score provides information pertaining to the situation described in section 4.4.2.

5.4 STATUS DATUM

5.4.1 STATUS

The status condition tag indicates if a concept´s match process is in any of three states: original, partial, complete. The original state exists before the match process begins. The partial state exists while the match process is ongoing, but not complete. The match status is in the complete state after all roles have been processed.

5.4.2 SUPER_MATCH

The super match datum is a pointer back to the owning pattern concept, whose role´s V/R link pointed to the concept. This datum is entered in the Match-Task at the time a pattern and target value description are put in correspondence and are about to be matched.

5.4.3 SUPER_ROLE

This is a pointer back to the owning role whose V/R link points to the concept.

5.4.4 TARGET

This is the target concept against which the pattern concept is matched for the Match Task.

5.4.5 ALIGNMENT TABLE

The role alignment table is stored as match datum if the match status is partial. Presently, the match process may not be suspended in the middle of the alignment process, so only fully determined alignment tables are stored.

5.4.6 OBLIGATORY MATCH COMPLETE

This flag indicates whether all obligatory roles have completed their matching.

5.4.7 EXAMPLE OF MATCH SCORING PROCESS

Figure (14) shows successive partial match scores as concepts of figure (7) are matched: a target description, the RENAME command, is matched against a pattern description, the COPY command. The four match scores change as successive termination object match results are triggered to the top level COPY concept. The main steps involved in this match are:
1) The pattern Match-Task scores are initially:
M[lb, base, estimate, ub] = M[0, 0, 0, 1].
The pattern COPY command´s roles are ordered and the target RENAME roles are aligned:

PATTERN ROLES:	Create*(.6),	Com-Word*(.3),	Syntax(.1)
TARGET ALIGN TYPE:	GEN	GEN	SUB

- The value restrictions of the Create roles of the pattern and target concepts are matched. Again there is a role ordering and alignment:

PATTERN ROLES:	Body*(.2),	Location(.4)	Name(.4),
TARGET ALIGN TYPE:	EQ	UNALIGNED	SUB

2) The Body roles are equal so a score is triggered upwards: (.2)(.6) = 1.2 . Match-Task scores are attached to the pattern concept:
M[0, 0, .12, 1]
The lower bound and base are not increased because the minimum score of .6 for the V/R has not been exceeded. All obligatory roles of the V/R , the single Body role , have completed their matching at this point.
3) The Location pattern role´s Location role is subsumed by the target role so there is a match failure resulting in a decrease of the upper bound:
M[0, 0, ,.12 ,.76].
4) The File Name concept subsumes the New File Name concept: (.4)(.6) + .12 = .36 and the top-level scores are updated to:
M[0, .36, .36, .76].
The base is incremented because the minimum required role score = the V/R´s upper bound = .6 . The lower bound remains at 0 because all obligatory roles of the COPY-Command have not yet been matched.
5) The Com-Word role value restriction is successfully generalization matched:
M[.66, .66, .66, .76].
The score and base are increased and now, since all necessary conditions are satisfied, the lower bound is incremented as well.
- The target role value restriction for the Syntax role is unknown, so the scores stay the same.

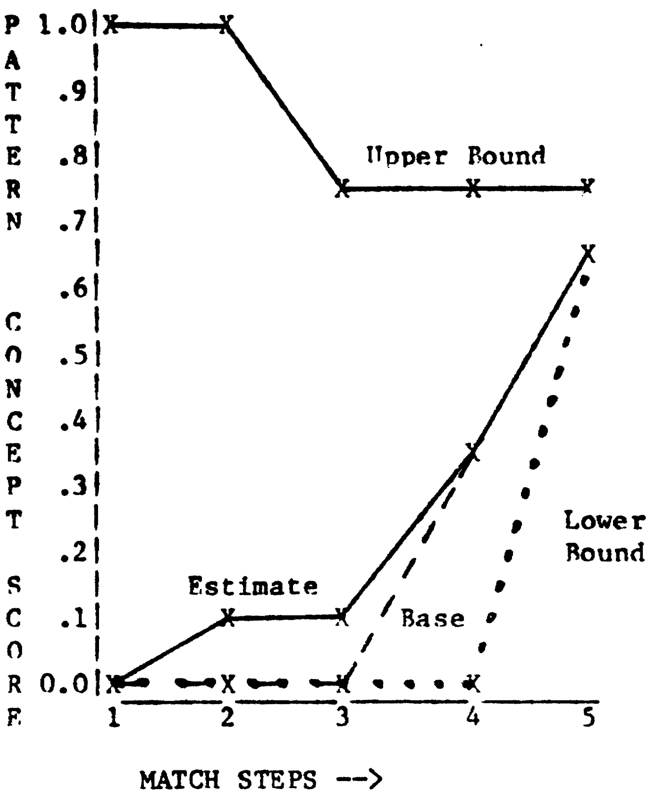


Figure (14) Incremental Match Scores of COPY command Description as the Target RENAME description is Matched

CHAPTER 6

INCREMENTAL SCORE UPDATES THROUGH TRIGGERING

Score triggering is the means by which a termination object's match score results are transmitted to its higher level concepts . In this way the original pattern concept's scores are the current best estimate based on all known match results at lower levels.

Score information is attached as part of a pattern concept's metadescription Match-Task whenever scores are updated. This score information is accessed as the trigger signal proceeds upwards. A successful match at the net's bottom level causes a score change of a role's value description. The score change is multiplied by the role's weight and the result is added to the owning concept's score. This process proceeds upwards until the original pattern concept is reached.

If a match fails, upper bounds are decremented, possibly causing role match failures at upper levels. If a role failure does occur, a record is kept of the proper next role for processing when the trigger signal is completed.

Figure (15) shows the relationships of the scoring and trigger objects. A sub-concept V/R is matched at the bottom of the network, and the score of its owning SuperC concept is updated. Scores are triggered to any Triggered Concepts at recursively higher levels, until the top level root pattern concept is reached.

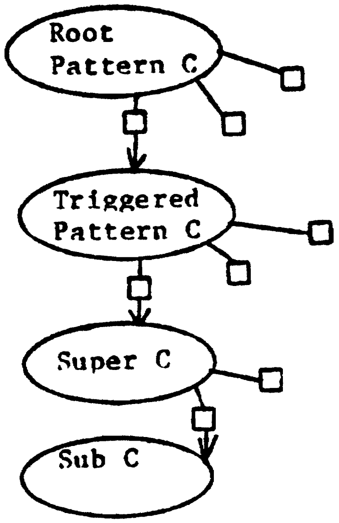


Figure (15) Trigger Chain

6.1 TRIGGER CONDITIONS

The following termination object matches result in the pattern concept´s match status being changed to complete, since final match scores have been determined.

6.1.1 CONCEPTS WITHOUT ROLES

If an equality, subsumption, or generalization relation is found between the pattern and target concepts without roles, the pattern concept is given an exact score of 1, which is triggered upwards. Otherwise a match failure score of 0 is triggered upwards.

6.1.2 UNKNOWN CONCEPTS

Unknown concepts do not change any scores so no triggering is performed.

6.1.3 EQUAL CONCEPTS

There are two cases to consider here. If both concepts are UNKNOWN then the procedure described in the previous section is used. Otherwise an exact match is scored and triggered upwards. The equal concepts could each contain exactly the same roles and match in this way. The concepts are necessarily exactly the same object for equality matching to succeed.

6.2 TRIGGERING DURING BEST SLOT ALIGNMENT

When multiple target roles align with a single pattern role, the best aligned target role is found by finding the best scoring target role. A problem arises if the usual triggering process is used since more than one role would trigger through the same role/V/R path.

To handle this case, the trigger signal terminates when it reaches the multiply aligned pattern role´s V/R instead of the top-level pattern concept. If N Match-Tasks are created at the lower level, N scores are compared to find the best match. The highest score is then triggered to levels above the "alignment" pattern concept level.

6.3 PROCESSING FLOW UPON FAILURE

During a failed match score triggering process the upper bound of a V/R concept may fall below the minimum role score of the role that points to that V/R. Figure (16) shows how an owning role's minimum score requirement serves as a threshold for failure as the V/R concept's roles are matched. As soon as role R2 fails, the owning role's score fails as well.

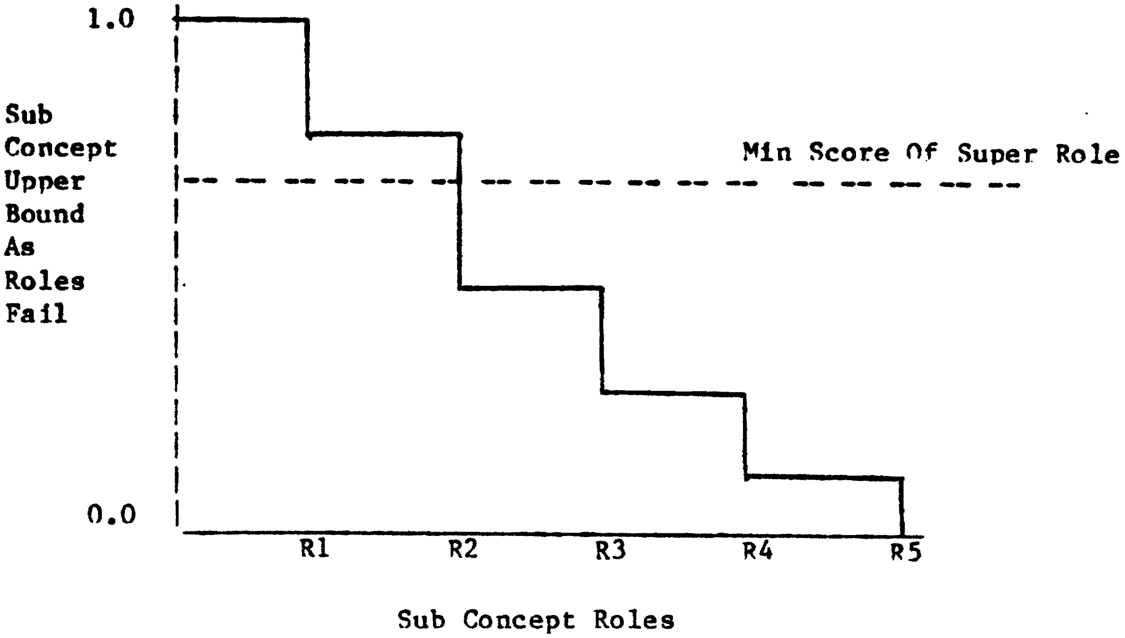
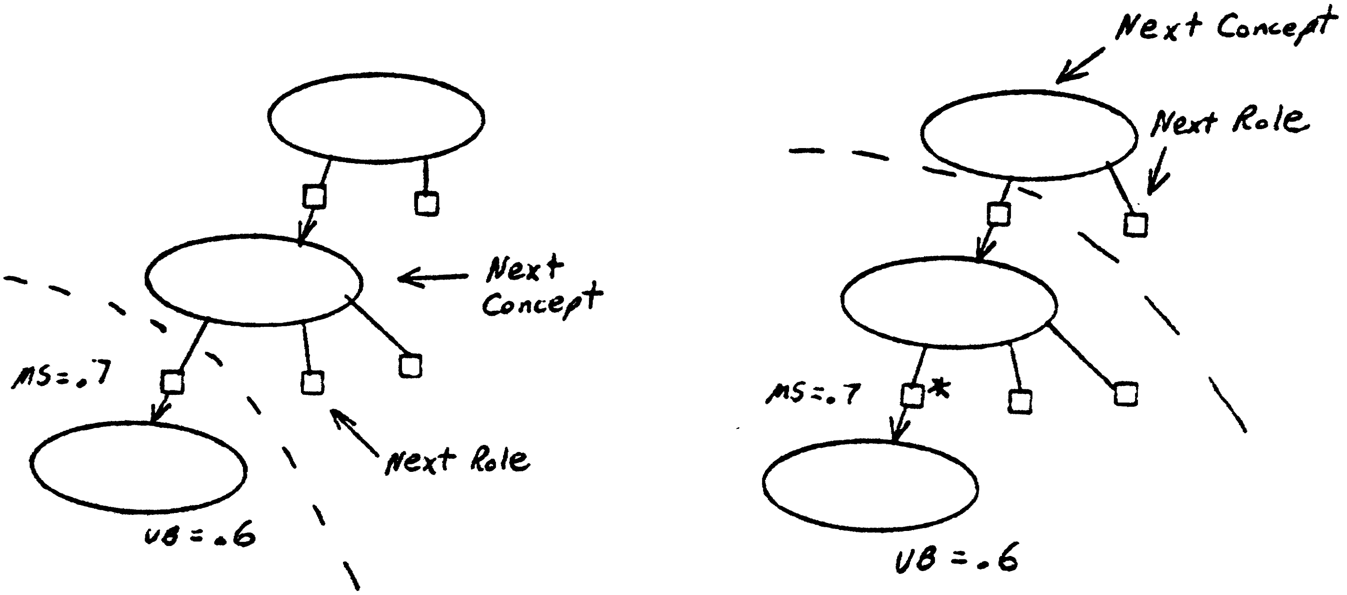


Figure (16) Minimum Score Threshold

This role failure could occur many levels above the termination object from which the trigger signal emanated. If the failed role's modality is optional, the owning concept's score must be reduced by the current role score. If the failed role's modality is obligatory, the owning concept's score is reset to 0, since the entire match fails when an obligatory role match fails.

After a score triggering process, flow of control would normally return to the role following the termination role on the ranked role list, or the next role on the super-concept's ranked role list if the sub-concept had just completed matching all roles on its ranked role list. If the preceding score triggering caused a role failure at a higher level, this is not the appropriate processing flow. The next role to be processed should be based on higher-level role failure requirements, if there were any. If a failure involving an optional role that was not the last role of its owning concept's ranked role list occurred, next_concept is set to the owning concept. If an obligatory role or the last role on the owning concept's ranked role list failed, next_concept is set to be the next higher concept. Figure (17) illustrates these actions.



Optional Role Failure

Obligatory Role Failure

Figure (17) Processing Flow Upon Failure

The bottom-level checking process is implemented by examining a global variable, ´triggeredfailure. If a triggered failure did occur, then a global variable next_concept would have been re-assigned to be the super-concept from which processing should proceed. The V/R match routine makes sure it is operating at the proper level. If not, exits are made until the recursive process pops up to the desired next_concept level. The appropriate next role will then be processed as normal cycling through the ranked role list occurs.

Concepts between the termination object and the concept owning the failed role will be left with a match status of complete so the restart procedure will not try to rematch them.

CHAPTER 7

MATCH SUSPENSION AND RESTART

7.1 INTRODUCTION

There are a number of situations involving recognition and interpretation of descriptions where it is advantageous to be able to suspend processing and restart at a later time. Suspension of the match before completion is useful if there are time constraints applicable or if one is willing to accept the match as soon as it reaches a minimum acceptable score level. It is conceivable that state changes in the problem domain may occur during the match process, for example changing a role´s modality from obligatory to optional. It may then be advantageous to immediately reorder the role match sequence and suspend processing on the present role because its priority has been reduced.

By retaining information describing previous partial match progress, suspended matches may be restarted without repeating the work the matcher previously performed.

7.2 MATCH SUSPENSION CONDITIONS

Match suspension may be specified to occur if the pattern´s score exceeds a threshold or if a time limit has been exceeded. The score threshold and time limit conditions are checked whenever score triggering reaches the pattern concept.

7.3 STORED DATUM

All required pattern match score and status information is continually updated and reattached whenever score changes are triggered through the network. In this way the pattern concepts retain information such that they are always "ready" for a match suspension. It is assumed that the match process may be suspended any time after an alignment table has been determined. In addition, if there is more than one target role that aligns with a pattern role, it is assumed that the best alignment has been found before suspension takes place. The implemented matcher examines all possible target/pattern role pairs to find the highest match score. Other strategies would be to quit processing as soon as one target/pattern match result is obtained or to quit processing immediately and assume no attempts have been made for that role´s match.

Other status and partial result information is stored when a match is suspended. The alignment table is stored, so it need not be recalculated if the match is restarted. As roles are value matched, they are removed from the alignment table. In this way the table serves as a record of remaining roles to value match. A three state status indicator is stored, to show if the match is in its original, partial, or completed state. Each Match-Task is tagged with a pointer to the target concept.

7.4 RESTART PROCEDURE

The restart procedure is a call to match the same pattern and target concepts that were earlier suspended. The pattern concept´s attached datum is searched for a Match-Task with the same target concept pointer. If none exists, a new Match-Task is created and score and status information is initialized.

If the desired target tagged Match Task is found, that Match-Task´s status is examined to see if it is partial or complete. Figure (18) is an example of the statuses of pattern concepts in a network after suspension and before restart has begun.

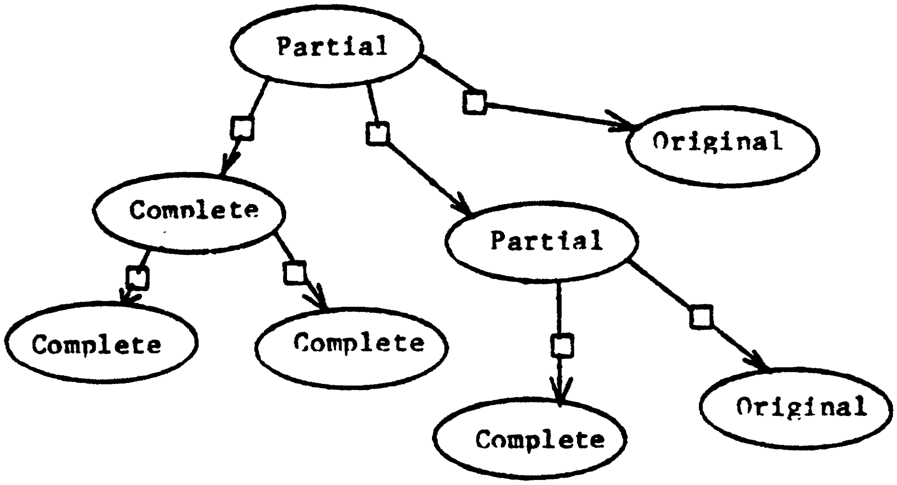


Figure (18) Example Status Settings for a Network to be Restart Matched

If the status is complete, the scores stored in that Match-Task have already been transmitted to the root pattern and recursive matching for that pattern/target pair need not be performed. If the status is partial, then a tree search is initiated to find the previous point of match suspension. The following algorithm describes the search process.

```
Find_Suspension_Point:

BEGIN

Find the first role on the ranked role list whose V/R´s
status is not ´complete

IF the status is ´original THEN
    return that concept as the
    suspension point
ELSE
    Find_Suspension_Point of the concept
    , its status being ´partial
END IF

END
```

When the restart role is found, the match proceeds as if it had just reached that point in the processing.

There are ways to make the restart position easier to establish. One would be to store the suspension point concept and role as part of the top-level pattern concept´s Match-Task when a suspension takes place. This would require passing of origination pointers with score trigger signals.

CHAPTER 8

CONCLUSION

8.1 SUMMARY

This thesis has described a description matcher for the KL-ONE knowledge representation language that is incremental and can perform partial matches. Matches may be performed incrementally by suspending a match process, based on score or time thresholds, and storing the necessary status and score information for a later match restart. The matcher is partial in that it has a formalism for measuring the degree to which two descriptions match. Lower bound and upper bound scores are seen to be analogous to the propositional support measures of Shafer's theory. The partial score capability and incremental nature of the matcher complement each other. Four scores determine the degree and confidence of a suspended match and the partial scores may be observed to be incrementally updated as new match information is found.

The match process for an example network has been explained and four scores compared graphically. The information provided by the match scores could provide useful information for analogical reasoning and general description comparison and recognition systems.

Particular contributions of this thesis include:

1. A more detailed examination of the scoring and triggering processes described in the Holmes paper, including extensions such as a minimum role score requirement, generalization matching, and a redefinition of the base score.

2. Determination of the match process implications with respect to the KL-ONE knowledge representation language.

3. Implementation ðf the partial matcher in Franz Lisp, using translated Franz Lisp KL-ONE functions.

8.2 FUTURE WORK

The matcher's power could be extended by returning more detailed match information. For example, the relative degrees of equality, subsumption, and generalization involved in a match is of interest. A distance measure could be used to show differences between subsumption depths. The match process could be extended to handle KL-ONE structural descriptions. It would also be useful to combine a weight adjustment and generalization mechanism to adaptively learn a best description of target descriptions.

The match process is assumed to be a top-down one where pattern descriptions guide the order of role comparison. Instead, a bottom-up recognition process could proceed through a "spreading activation" (Quillian, 1968) from input descriptions. Higher level concepts with highest partial scores could then be recognized. The following sections provide more detail for possible matcher extensions.

8.2.1 ADDITIONAL SCORE INFORMATION

8.2.1.1 Distance Measure -

Semantic distance measures have been considered by psychologists (Tourangeau,1978) as a means of measuring the similarity between concepts. Tourangeau suggests that "differences in salience" may be factored into the formulation with an equation such as:

EQUATION (1):
Dk(A,B) = [(sum i to m): Wik(a(i) - b(i))**2] **.5 ,

where Wik is the relative weight of feature i in context k. Tversky suggests a distance measure that distinguishes between features unique to and shared between the concepts being compared. His equation is:

EQUATION (2):
D(A,B) = b1*f(a-b) + b2*f(b-a) + b3*f(a INT b) ,

where a-b are features unique to a, b-a are features unique to b, and a INT b are features a and b have in common.

For present purposes, we are interested in the meaning and usefulness of a distance measure related to the matching task: measure the degree to which one concept may be viewed as another concept. Since match scores are not reflexive with respect to the pattern and concept SuperC positions, we expect the distance measure to be irreflexive as well. We are interested in distances between roles of the pattern concept and aligned target roles. The following equation can be used:

EQUATION (3):
d(P,T) = (sum over pattern roles):
ABS[W(Rpi) - (SCORE(Rpi to Rti))*W(Rti)] / 2

where ABS = absolute value and SCORE is the partial match score of the role V/Rs. Rpi is the ith pattern role , Rti is the ith target role, and d(P,T) is the distance from the pattern to the target concept. The maximum value of this distance is 1 if the roles sum to 1 and there are two or more roles. Assume that all concepts with roles will have at least two roles, so a type of normalization is enforced. If the pattern and target concepts do not have roles themselves, this equation is not used to calculate a distance measure.

By multiplying the target's weight by the pattern to target role score a salience with respect to the pattern role is formed. This allows comparison with the pattern role's weight. One might consider using an equation with a term of the form:
SCORE(Rpi to Rti) * [W(Rpi) - W(Rti)] ,

but note what happens when the score is 0: the distance goes to 0 as well. It is more reasonable to contribute a distance equal to W(Rpi) in this case, so the original equation is used. If the score term is eliminated completely, the partial match conditions are not fully factored into the distance measure.

Figure (19) shows an example general network in which a distance will be measured. Figure (20) shows a "Venn Salience" diagram as a means of obtaining a visual grasp of the differences in role saliences between two concepts. Area coverage is used to represent weight magnitude. For example, role R1 of C1 covers one quarter of C1's area but only one tenth of the area of the entire C2 concept. It is this difference in salience area that is being captured in the distance measure. To understand how the SCORE term of the distance equation is factored into the distance measure, consider the role areas of R1 and R2 of C2 as possibly shrinking to 0, depending on the match scores. Then the R1 area could cover an area from 0 to .1 and the R2 area could cover an area from 0 to .3.

The distance between C1 and C2, using equation (3), and assuming that the V/R match scores of C1 and C2 are .5 , is:

$$d(C1,C2) = [\ |\ .25 - (.5)(.1)\ | + |\ .75 - (.5)(.3)\ |\ ]/2 =$$
$$[.2 + .6]/2 = .4\ .$$

Note that if the SCORE term were not used, or perfect matches were assumed , the distance would be:

$$d(C1,C2) = [\ |\ .25 - (.1)\ | + |\ .75 - (.3)\ |\ ]/2 =$$
$$[.15 + .45]/2 = .3\ .$$

The fact that only a partial match exists increases the distance between the concepts.

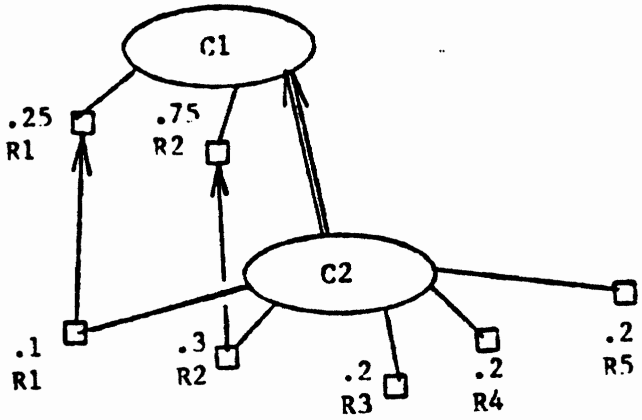


Figure (19) Example Net For Distance Measure

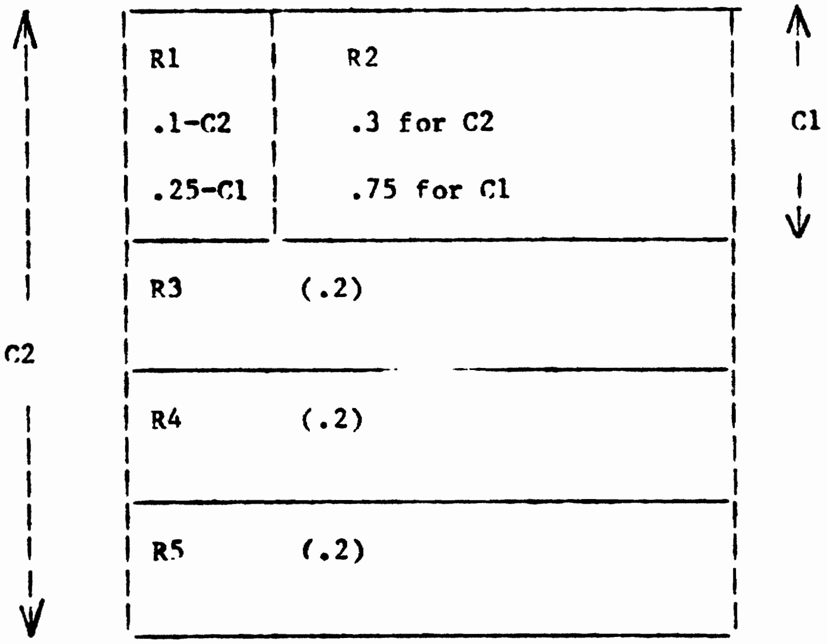


Figure (20) Venn Salience Diagram For Figure (19)

Figure (21) shows a bird concept network that might exist as an abstraction in someone's mind. Assuming that perfect subsumption matches occur between the BIRD and EAGLE and BIRD and PENGUIN concepts, this example shows that additional information is added by a distance measure.

The BIRD to EAGLE distance is:

$$[\ |.2 - .2| + |.5 - .4| + |.1 - .1| + |.1 - .1| + |.1 - .1|\ ]/2 =$$
$$[\ 0 + .1 + 0 + 0 + 0\ ]/2 = .05\ .$$

The BIRD to PENGUIN distance is:

$$[\,|.2 - .05| + |.5 - .05| + |.1 - .4| + |.1 - .3| + |.1 - .2|\,]/2 =$$
$$[\ .15 + .45 + .3 + .2 + .1\ ]/2 = .6\ .$$

If the distance measure were not used in this case both the EAGLE and PENGUIN concepts would be perfectly subsumed by the BIRD concept. By using the role weight information of the target concepts, the proposed distance measure shows that an EAGLE is closer in meaning to the BIRD concept than the PENGUIN concept is.

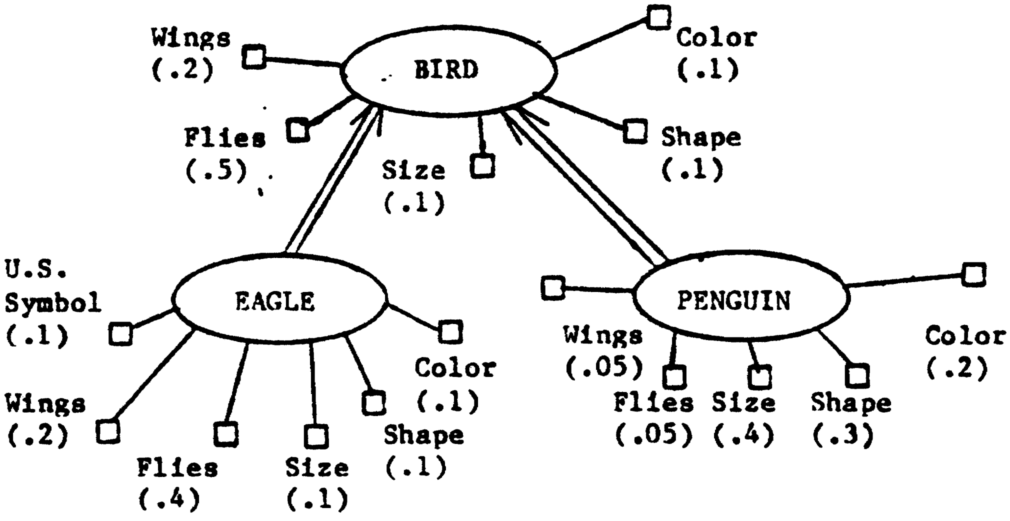


Figure-(21)-Bird-concept-network

8.2.1.2 Association Lists -

Data associations found in the process of matching a target concept to a pattern concept could be returned. For example, if a concept marked ? is used as a wildcard concept , the concept it matches could be returned.

Generalization matches allow flexibility of abstraction reasoning to the matcher. However, the reasoning process may need to know what generalizations were required to perform the match. This information could be accumulated whenever a pattern concept is generalized to a higher level concept.

If the reasoning process is to perform intelligent backtracking or is to attempt to learn from "mistakes", a list of objects that failed to match could be useful.

8.2.1.3 Number -

The Number is expressed in terms of the pair <min max> . Consider equality, subsumption, and generalization matches:

EQUALITY

This condition is satisfied if

min(pattern) = min(target)
max(pattern) = max(target)

e.g

Pattern	Target
(1 5)	(1 5)
(2 2)	(2 2)

SUBSUMPTION

min(pattern) <= min(target)

max(pattern) >= max(target)

e.g

Pattern	Target
(1 5)	(2 4)
(1 5)	(3 3)

GENERALIZATION

A role that is an ancestor of both the pattern and target roles should have a range that includes the ranges of both the pattern and target Numbers:

min(gen(pattern)) <= min(target)
min(gen(pattern)) <= min(pattern)
max(gen(pattern)) >= max(target)
max(gen(pattern)) >= max(pattern)

8.2.2 ALTERNATIVE MATCH PROCEDURES

One may put constraints on the recursive match process described here and construct a more sensitive matcher. For example, when first describing the pattern concepts, one may disallow generalization role alignments and generalization concept matches. A distance threshold could be added as a role subsumption requirement. Another possibility is that one could disallow generalizations if the ancestor concept is hierarchically above a "basic" concept level.

8.2.2.1 Alternative Termination Object Scoring -
The implemented matcher returns exact match scores of 1 or 0 when termination objects are reached. Another possibility is to specify that a measurement will be taken and the measurement score value, a number from 0 to 1, will be returned as the score for the pattern concept. Quantitative data such as statistical measurements could then be factored into the match process. A robot system might interface its perceptual sensor readings with a semantic net representation of scene structure in this manner.

8.2.2.2 Root-Pattern-Tagged Match Tasks -
The Match-Tasks could also be tagged by the root pattern concept. this is necessary in a parallel processing environment where more than one root pattern Match-Task is ongoing. The various match processes may reach the same V/R pattern concept and unless the proper partial state of processing with respect to their own suspension state is used, improper scoring will occur. A datum ´last role matched could be used to keep track of the last role matched. In this way the alignment of roles would not need to be redone if the same pattern- target concept pair were to be matched by two different root pattern concepts´ match processes.

8.2.2.3 Match Queue -
The FFRL matcher used a match strategy called "fm-recall-result", which checked a queue of previously stored match results as a first step in the match attempt (Finin,1980,p80). A maximum queue size was maintained and old match results were dropped off the bottom when the queue became full. The presented partial matcher attaches tagged Match-Tasks to pattern concepts. If these were stored in a queue, a similar approach could be used.

8.2.2.4 Weighting -
A role´s weight can be thought of as applicable to the conjunction of the functional match as well as filler match aspects of the role. One could decompose a role´s weight into its functional and filler components. For example, if a concept has a role with weight w(role) , let

w(functional) + w(filler) = 1

and a role score is determined by:

w(role) * [w(funct.)*score(funct.) + w(fill.)*score(fill.)] .

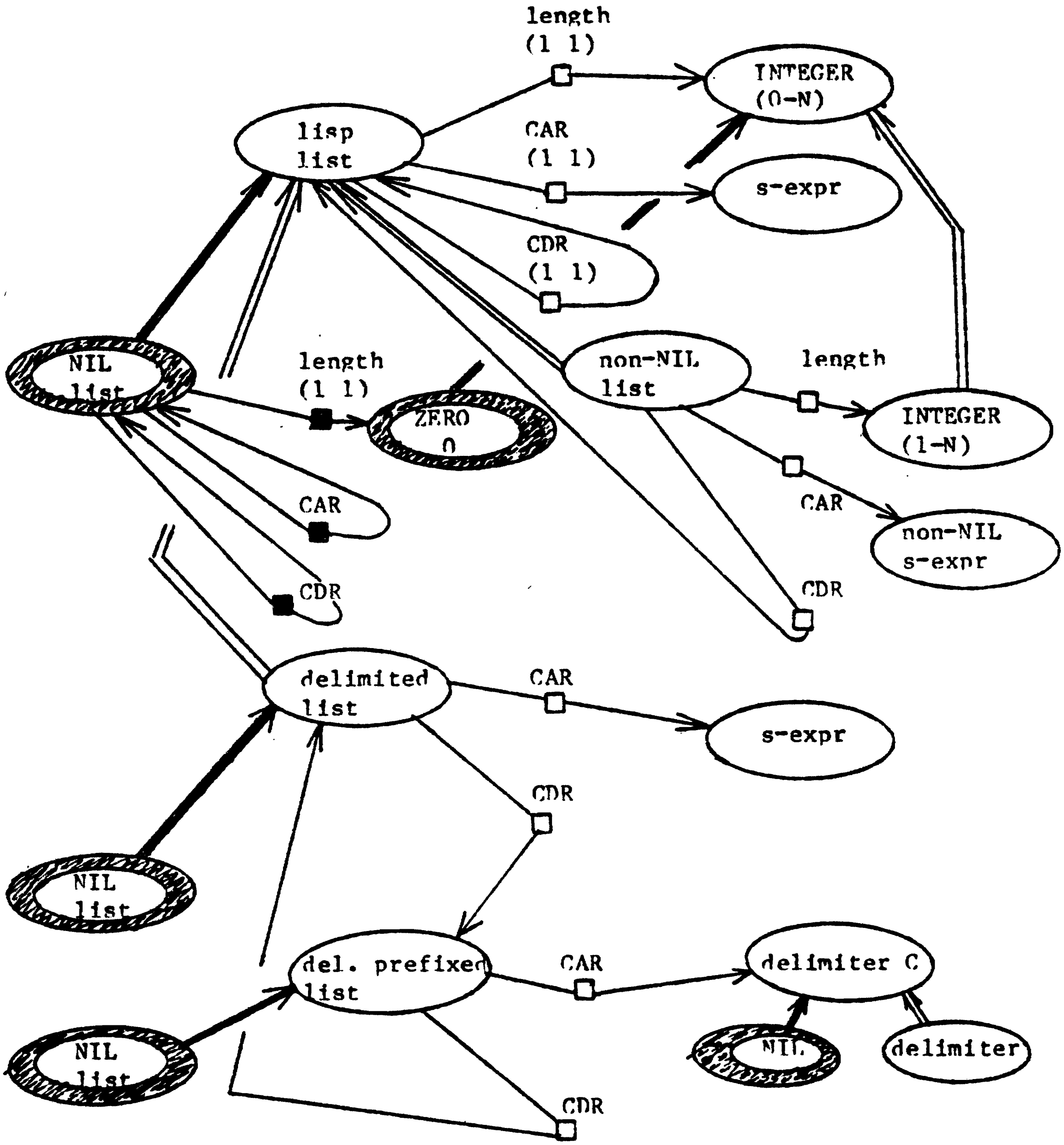
8.2.2.5 Structural Descriptions -

Structural descriptions serve to place constraints on role fillers so that the functional meaning of roles is described. Since the structural description adds to the concept definition in a way similar to the role description facility, weights could be attached to the structural descriptions for matching purposes. Structural description matching could be modeled after the partial match scheme described in this thesis if the pattern and target paraindividuated concepts are matched. A best SD match could serve as a criteria for choosing a best generalization alignment path when multiple paths exist.

8.2.3 CYCLE HANDLING

Although Generic concepts may not form cycles through SuperC arcs, recursive definitions may be constructed by allowing a concept´s role´s value restriction be the concept itself. For example, in figure (22), lisp-list´s cdr role has the lisp-list as its V/R.

Less direct cycles are formed whenever the value restriction of a role of a sub-concept reached through the V/R links describing a main concept points to another concept that is part of the main concept´s descriptional network. For example, in figure (22), delimiter-prefixed-list links back to delimited-list; the concepts serve as V/Rs for each other. The pattern network may be cycled through recursively as long as the same target concept tagged Match-Task is not visited twice.



EXAMPLES:
    Lisp List: () , (a b)
    Delimited List: (a , b)

Figure (22) Lisp list representation

Unless precautions are taken, the matcher could get caught in an infinite loop if presented with a pattern and target concept that are both recursively defined. For example, if a lisp list is matched against itself, an identical concept match is attempted when the V/Rs of the cdr roles are to be matched. This case is not handled by the implemented matcher. The matcher can not simply avoid a match attempt, because the match should proceed if the concepts are partially matched by way of a previous match suspension. A "previously reached" mark could be attached to each pattern concept reached during a single match attempt. If a previously reached concept is to be rematched, the matcher would detect a cycle would be entered and would quit the match process. All "previously reached" marks must be removed after a single match processing task. An alternative procedure could be to maintain a dynamic stack, as was done in the FFRL matcher.

CHAPTER 9

BIBLIOGRAPHY

1. Bobrow, D.G. and Winograd, T.W.; (1977a);
   An Overview of KRL-0, a Knowledge Representation Language;
   Cognitive Science, Volume 1, Number 1

2. Bobrow, D.G., Winograd, T.W.,
   and the KRL research group; (1977b);
   Experience with KRL-0 One Cycle of a Knowledge
   Representation Language;
   Proceedings of the 5th International Joint Conference
   on Artificial Intelligence; pages 213-222

3. Brachman, R.J.; (April 1978a);
   Theoretical Studies in Natural Language Understanding;
   BBN Report #3888; Bolt Beranek and Newman

4. Brachman, R.J.; (May 1978b);
   A Structural Paradigm for Representing Knowledge;
   BBN Report #3605; Bolt Beranek and Newman

5. Brachman, R.J., Ciccarelli E., Greenfeld N.,Yonke M.;
   (July 1978); KLONE Reference Manual;
   BBN Report #3848; Bolt Beranek and Newman

6. Brachman, R.J.;(1979a);
   On The Epistemological Status of Semantic Networks;
   Associative Networks; N. Findler (Ed.); Academic Press, Inc.

7. Brachman, R.J., Bobrow, R.J., Cohen, P.R.,
   Klovstad, J.W., Webber, B.L., Woods, W.A. ;(1979b);
   Research in Natural Language Understanding;
   BBN Report #4274; Bolt Beranek and Newman

8. Evans, T.D. (1968);
   A Program for the Solution of a Class of Geometric-Analogy
   Intelligence-Test Questions; in Semantic Information Processing;
   M. Minsky (Ed.); The MIT Press

9. Fahlman, S.E.; (1979);
NETL: A System for Representing an Using Real-World Knowledge;
The MIT Press

10. Finin, T.W.;(June, 1980);
The Semantic Interpretation of Compound Nominals ; PH.D. Thesis

11. Finin, T.W.;(1982);
Translating KL-ONE from Interlisp to Franzlisp;
Proceedings 2cd KL-ONE Workshop

12. Finin, T.W.,Webber, B.L.;(1983);
BUP - A Bottom Up Parser; University of PA;
Technical report MS-CIS-1983-16

13. Garvey, T.D. , Lowrance, J.D. , Fischler, M.A.; (1981);
An Inference Technique for Integrating Knowledge from Disparate Sources;
Proceedings of the 7th International Joint Conference
on Artificial Intelligence; pages 319-325

14. Holmes, Peter D.; (May 1981);
Confronting the Matching Problem ;
Senior Project ; University of PA

15. Lipkis,Thomas; (1981);
A KL-ONE Classifier; USC/I.S.I.; Consul Note No. 5

16. Moore, J. and Newell, A.; (1973); How Can MERLIN Understand;
Knowledge and Cognition, Hillsdale, N.J., L. Gregg (Ed.);
Lawrence Erlbaum Assoc.

17. Quillian, M.R.; (1968); Semantic Memory;
Semantic Information Processing; M. Minsky (Ed.); The MIT Press

18. Shafer, G.; (1976);
A Mathematical Theory of Evidence;
Princeton Universiy Press, Princeton, New Jersey

19. Tourangeau, Roger and Sternberg, Robert J.; (1978);
Understanding and Appreciating Metaphors;
Report No. 11; Department of Psychology, Yale University

20. Winston, P.H.; (1975) ;
Learning Structural Descriptions From Examples;
The Psychology of Computer Vision; P. Winston (Ed.);McGraw-Hill

21. Woods, W.A. , (1975);What's In a Concept;
Representation and Understanding;

22. Woods, W.A. , (1978);
Taxonomic Lattice Structures for Situation Recognition;
TINLAP-2, University of Illinois at Urbana-Champaign

23. Woods, W.A. , (February 1979); Research in Natural Language Understanding; Report #4181; Bolt Beranek and Newman

APPENDIX A

SHAFER REPRESENTATION

Work by Garvey used Shafer´s mathematical theory of evidence as a basis for an "evidential propositional calculus". In that formalism a proposition´s likelihood is represented as an interval [s(A),p(A)], where s(A) is the evidential support for proposition A and p(A) is the degree of plausability for A. s(A) and p(A) are similar to the lower and upper bound score measures , [lb,ub], used in this matcher. Figure (23) shows the relationships.

Shafer Theory		Partial Matcher	
Term	Meaning	Term	Meaning
A	proposition A	M	Match: The target concept can be viewed as the pattern concept.
s(a)	evidential support for A	lb(M)	concept match success for M
p(a)	1 - evidential support for ~A	ub(M)	1 - concept match failure for M

Figure (23) Comparison of Shafer and Partial Matcher measures

When the lower bound is not equal to the upper bound, a subinterval of uncertainty exists. Examples given in (Garvey, 1981) may be interpreted with respect to the lb and ub notation of this partial matcher, as shown in figure (24).

Shafer	Partial Matcher
A[0,1] no knowledge about A	M[0,1] initial state of match, no matching done yet
A[0,0] A is false	M[0,0] the target may not be seen as the pattern at all
A[1,1] A is true	M[1,1] the target may be seen as the pattern completely
A[.25,1] partial support for A	M[.25,1] the target is partially the pattern
A[0,.85] partial suoport for ~A	M[0,.85] the target is partially not the pattern
A[.25,.85] partial support for A and ~A	M[.25,.85] the target is partially the pattern and the target is partially not the pattern

Figure (24)Comparison of Shafer Interval
to Partial Match scores

The resultant lb and ub scores from the presented match method may be used as an input for the knowledge integration technique used by Garvey.

Garvey uses Shafer´s theory in conjunction with Dempster´s Rule of Combination to determine the most likely signal emitter given different types of sensor measurements. Two feature measures are used to try to determine which of five independent emitters is the most likely source of the observed signals.

Posed in the KL-ONE framework, five pattern emitter concepts would each be matched against the measured target emitter concept. The match score process described in earlier chapters would need to be supplemented with a Dempster´s rule convention in order to obtain the lower and upper bound scores of the Shafer-Dempster theory. One strategy would be to mark certain roles as "identifying" roles. The concept´s scores would determined by using Dempster´s theory directly on the identifying role lower and upper bounds and those roles´ weights would not be factored into that calculation. This approach is suited to descriptions that become more certain as information is synthesized, rather than added together as separate decompositional descriptional units of a concept. The "identifying" roles could also be labeled "wholistic" roles. Supporting evidence from those roles is factored together and the score of their combination is greater than a sum of scores from "decompositional" roles. One might treat the resultant lb and ub from all decompositional roles as one wholistic role and factor that resultant in with other wholistic roles . That would allow combination of both types of information to achieve more accurate scores.

APPENDIX B

MATCH EXAMPLE

B.1 ENTRY METHOD

The main tool used to enter descriptions is a bottom-up parser, developed by Dr. Finin. A grammar, similar in some aspects to the JARGON language developed at BBN (Brachman,1979b) as well as the "A" language (Finin,1980,p61) was constructed to allow for natural entry of concepts, roles , and there relationships , as needed for the partial matcher networks. The steps followed to enter KL-ONE descriptions are:

1. Create a graphic diagram of the description, in terms of named concepts, named roles, role weights, role min scores, and arcs. Figure (7) is an example.

2. Starting with the most general concepts, create a natural language sentence describing the KL-ONE objects. These sentences are grouped into one list for BUP translation (Finin ,1983). Figure (25) lists some of the entry sentences used to build a network corresponding to Figure (7).

3. Run a program which iterates through the list of entry sentences, calls BUP, and adds the translation, which is in the form of a KL-ONE function call, to an output generation list.

4. Run a program which iterates though the generation list and causes evaluation of each KL-ONE function call.

5. Create formatted entry lists that assign weights and min scores to specified roles of concepts.

6. Run a program which iterates through the list of weights and min scores, forms a list of weights and min scores, and attaches them to the owning concept. A ranked role list is also determined and attached to the owning concept.

```
(a Command                       is a thing)
(a Com-Word                      is a thing)
(a Syntax                        is a thing)
(a Command                       must have 1 Word which
                                 is a Com-Word)
(a Command                       may have 1 Syntax which
                                 is a Syntax)
(a RENAME-Word                   is a Com-Word)
(a COPY-Word                     is a Com-Word)
(an Object-With-File-Name        is a thing)
(a  File-Name                    is a thing)
(an Object-With-File-Name        may have 1 Name which
                                 is a File-Name)
(an Object-With-Dir-Location     is a thing)
(a Directory                     is a thing)
(an Object-With-Dir-Location     may have 1 Location which
                                 is a Directory)
(a File-Body                     is a thing)
(a File                          is an Object-With-File-Name)
(File                            specializes
                                 Object-With-Dir-Location)
(a File                          must have 1 Body which
                                 is a File-Body)
(a Command-To-Create-File        is a Command)
(a Command-To-Create-File        must have 1 Created-File
                                 which is a File)
(a New-File-Name                 is a File-Name)
(a Renamed-File                  is a File)
(a Renamed-File                  may have 1 Name which
                                 is a New-File-Name)
(Renamed-File                    has a role Name
                                 which modifies
                                 the role Name
                                 of Object-With-File-Name)
(a Different-Directory           is a Directory)
(a File-Copied-To-New-Dir        is a File)
(a File-Copied-To-New-Dir        may have 1 Location which
                                 is a Different-Directory)
(File-Copied-To-New-Dir          has a role Location
                                 which modifies
                                 the role Location
                                 of Object-With-Dir-Location)
```

Figure (25) Example Entry Sentences

B.2 MATCH TRACE

Figure (26) shows a trace of the match described in section 5.4.7 . Role alignments and scoring updates are shown for each match task and subtask set-up in the match process. Whereas Figure (14) shows the match progress only at the highest root node, Figure (26) also shows the status and score changes at the lower level concept: File-Copied-To-New-Dir. Score changes at the termination object level are not shown. A time limit of 1000 cpu units was set for the match process to illustrate the suspension and restart capability.

```
                              BEGIN MATCHING PROCESS
                              ----------------------
THE TOP-LEVEL MATCH IS BETWEEN
THE PATTERN CONCEPT: (CONCEPT: COPY-Command)
THE TARGET  CONCEPT: (CONCEPT: RENAME-Command)

CONCEPT MATCH (CONCEPT: COPY-Command) and (CONCEPT: RENAME-Command)

                              ALIGNMENT TABLE
                              ---------------
   PATTERN ROLE              TARGET ROLE                 ALIGN TYPE
   ------------              -----------                 ----------
1  (Created-File TYPE:generic)(Created-File TYPE:generic)  generalization
2  (Word TYPE:generic)       (Word TYPE:generic)         generalization
3  (Syntax TYPE:generic)     (Syntax TYPE:generic)       subsumption

CONCEPT MATCH (CONCEPT: File-Copied-To-New-Dir) and (CONCEPT: Renamed-File)

                              ALIGNMENT TABLE
                              ---------------
   PATTERN ROLE              TARGET ROLE                 ALIGN TYPE
   ------------              -----------                 ----------
1  (Body TYPE:generic)       (Body TYPE:generic)         equality
2  (Location TYPE:generic)   unaligned                   unaligned
3  (Name TYPE:generic)       (Name TYPE:generic)         subsumption
SUCCESS: EQUAL V/Rs: File-Body

                              MATCH-TASK STATUS AND SCORES
Pattern:                (CONCEPT: File-Copied-To-New-Dir)
Target:                 (CONCEPT: Renamed-File)
Obl roles complete      t
Super_match             (CONCEPT: COPY-Command)
Super_role              (ROLE: (Created-File TYPE:generic))
[     lb   base est     ub    ]
[     0.2  0.2  0.2     1     ]

TRIGGER UPWARDS
                              MATCH-TASK STATUS AND SCORES
Pattern:                (CONCEPT: COPY-Command)
Target:                 (CONCEPT: RENAME-Command)
```

```
Obl roles complete  nil
Super_match         nil
Super_role          nil
[     lb    base est  ub    ]
[     0     0    0.12 1     ]

Current time: 830
FAILURE UNALIGNED PATTERN ROLE: (Location)

                          MATCH-TASK STATUS AND SCORES
Pattern:            (CONCEPT: File-Copied-To-New-Dir)
Target:             (CONCEPT: Renamed-File)
Obl roles complete  t
Super_match         (CONCEPT: COPY-Command)
Super_role          (ROLE: (Created-File TYPE:generic))
[     lb    base est  ub    ]
[     0.2   0.2  0.2  0.6   ]

TRIGGER UPWARDS
                          MATCH-TASK STATUS AND SCORES
Pattern:            (CONCEPT: COPY-Command)
Target:             (CONCEPT: RENAME-Command)
Obl roles complete  nil
Super_match         nil
Super_role          nil
[     lb    base est  ub    ]
[     0     0    0.12 0.76  ]

Current time: 1060
THE MATCH PROCESS HAS BEEN SUSPENDED
After this match process the match task data is:

                          MATCH-TASK STATUS AND SCORES
Pattern:            (CONCEPT: COPY-Command)
Target:             (CONCEPT: RENAME-Command)
Obl roles complete  nil
Super_match         nil
Super_role          nil
[     lb    base est  ub    ]
[     0     0    0.12 0.76  ]


                          BEGIN MATCHING PROCESS
                          ----------------------
THE TOP-LEVEL MATCH IS BETWEEN
THE PATTERN CONCEPT: (CONCEPT: COPY-Command)
THE TARGET  CONCEPT: (CONCEPT: RENAME-Command)

CONCEPT MATCH (CONCEPT: COPY-Command) and (CONCEPT: RENAME-Command)
The match was partially completed and will be continued
```

```
CONCEPT MATCH (CONCEPT: File-Copied-To-New-Dir) and (CONCEPT: Renamed-File)
The match was partially completed and will be continued

CONCEPT MATCH (CONCEPT: File-Name) and (CONCEPT: New-File-Name)

No roles, so do role-less concept match
SUCCESS: File-Name SUBSUMES New-File-Name

                         MATCH-TASK STATUS AND SCORES
Pattern:            (CONCEPT: File-Copied-To-New-Dir)
Target:             (CONCEPT: Renamed-File)
Obl roles complete  t
Super_match         (CONCEPT: COPY-Command)
Super_role          (ROLE: (Created-File TYPE:generic))
[    lb   base est  ub   ]
[    0.6  0.6  0.6  0.6  ]

TRIGGER UPWARDS
                         MATCH-TASK STATUS AND SCORES
Pattern:            (CONCEPT: COPY-Command)
Target:             (CONCEPT: RENAME-Command)
Obl roles complete  nil
Super_match         nil
Super_role          nil
[    lb   base est  ub   ]
[    0    0.36 0.36 0.76 ]

Current time: 619

CONCEPT MATCH (CONCEPT: COPY-Word) and (CONCEPT: RENAME-Word)

No roles, so do role-less concept match
SUCCESS: COPY-Word GENERALIZES TO SUBSUME RENAME-Word

                         MATCH-TASK STATUS AND SCORES
Pattern:            (CONCEPT: COPY-Command)
Target:             (CONCEPT: RENAME-Command)
Obl roles complete  t
Super_match         nil
Super_role          nil
[    lb   base est  ub   ]
[    0.66 0.66 0.66 0.76 ]

Current time: 1059
An UNKNOWN V/R was found
After this match process the match task data is:

                         MATCH-TASK STATUS AND SCORES
Pattern:            (CONCEPT: COPY-Command)
Target:             (CONCEPT: RENAME-Command)
Obl roles complete  t
Super_match         nil
```

```
Super_role          nil
[     lb   base est  ub    ]
[     0.66 0.66 0.66 0.76 ]
```

Figure (26) Example Match Trace