ON SEMANTIC NETS, FRAMES AND ASSOCIATIONS

Philip J. Hayes
Department of Computer Science
The University of Rochester
Rochester, NY 14627, U.S.A.

Abstract

A knowledge representation system is presented, based on the use of a semantic net on which a higher level structure of frames has been superimposed. The system was designed for use with a natural language system for finding the correct senses of ambiguous words in context. An examination of several linguistic examples shows how the representation system facilitates associative searches of context for potentially appropriate senses of ambiguous words, and how the results of such searches can often provide definite referents. The system also embodies novel approaches to the representation of multiple subparts, and of similar, but different, entities.

I Introduction

Recently, there have been several attempts ([1], [2], and [4]) to make semantic nets more useful as a method of knowledge representation. All these projects have depended on use of frame-like structures ([5]); that is, they introduced a unit of representation that is on a higher level than the nodes and links of more traditional semantic nets ([6], [7]), in that it stands for the information contained in a collection of several nodes and links.

These frame-like structures allow a notion of relevance or viewpoint to be implemented. In any given use of the semantic net, only some of the frames are deemed active, and the information contained in the others is ignored; in this way, the information apparently contained in the net can be made dependent on the point of view of the process accessing the net. As the several reports mentioned above show, this basic technique can be used very effectively to a number of different purposes, including the representation of quantification, of variable levels of detail, of contradictory hypotheses, and of different aspects of the same object.

This paper describes the application of the technique to a more specialised problem, that of word-sense disambiguation by association. By an association, I will mean some definite and systematic relation between two concepts; for example, in:

When Fred arrived, his arm was covered in bandages.

The woman went to the clock, and looked at the face.

The gambler looked at his hand.

the appropriate senses of "arm", "face", and "hand", respectively have associations with the appropriate senses of "man", "clock", and "gambler". Superimposing a frame-like structure on top of a large semantic net allows the number of search steps involved in finding such associations to be reduced; in addition, defining several different classes of relationships between frames, and making the search sensitive to them, allows the search to be controlled very tightly.

These associative techniques have been implemented in CSAW, a working computer system for choosing the .senses of ambiguous words in context. The associations they provide are used to find the appropriate senses and referents for ambiguous words in examples like those above. Of course, association, by itself, cannot resolve all ambiguous words, and CSAW, described in detail in [3], incorporates a number of other disambiguating techniques. What follows will, however, deal only with association.

Besides allowing associative searches to be tightly controlled, some of the relations between frames in the data base of CSAW are interesting from a more general representational point of view. In particular, the subset-superset relation allows significant economies in the representation of similar, but different, objects; while the part-whole relation incorporates some novel ideas on the representation of multiple subparts. These two relationships are of fundamental importance in the knowledge representation system of CSAW, providing a basic framework for the data base through the classes of orthogonal hierarchies that they generate.

II Depictions and Binders

Rather than use the word "frame", I have chosen to call the higher level units in my representation <u>depictions</u>. This both avoids tiring an already overworked word still further, and is more suggestive of the role such entities play. A depiction is a collection of nodes and links, which forms a subnet of a large semantic net. One of the nodes in a depiction is identified as the <u>depictee</u>; depictions describe their depictee in terms of their other nodes, which are known as <u>depicters</u>.

As an example, consider the very simple physical depiction of a human shown in Figure i. The solid node N-HUMAN is the depictee, while the hollow nodes are the depicters. The depiction, D-HUMAN, consists of all the nodes within the large closed dashed line, plus all the links between those nodes, plus both the links leading out of the enclosed area. The two nodes outside the dashed line are not in the depiction, even though they are part of the description of the depictee.

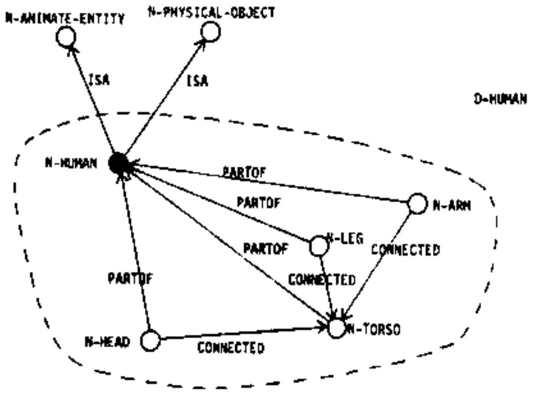

Figure i   A simple depiction of a human

This last observation is an important point because of the quantification implied by depictions. All nodes in D-HUMAN are generic nodes in the sense that they represent archetypal bodies, legs, torsos, etc. However, they cannot all be universally quantified, otherwise the representation would say, e.g., that all arms were connected to all bodies. Instead only the depictee is universally quantified, and all the depicters are existentially quantified within the scope of that universal quantifier.

While depictions are generic, they can be instantiated to produce descriptions of particular objects. Depictions are always instantiated as a whole by a <u>binder</u>. A binder is a data structure containing, among other things, the name of the depiction it instantiates, and a list of correspondences between generic nodes in that depiction and <u>instance nodes</u>. Instance nodes represent specific rather than generic objects, and as their name implies, these specific objects are instances of the generic objects with which they are paired by their binder. Figure ii shows an instantiation of D-HUMAN. The binder, D-HUMAM1, is diagrammed by the square box, and the double-lined arrow from it to D-HUMAN shows that it is a binder of D-HUMAN. Fred, Fred's torso, and Fred's arm, are all instance nodes which are paired in D-HUMAN#1 with the generic nodes in D-HUMAN to which they are connected. The line intersecting these connexions in dots and pointing at D-HUMAN#1 indicates the presence of these pairings in D-HUMAN#1.

Instance nodes inherit properties from their generic nodes. In particular, they inherit the relations of their generic nodes with other nodes. Of course, the inherited relations are not with other generic nodes, but are rather with the appropriate instance node. Thus in Figure ii, Fred's arm inherits from D-HUMAN the relation not of being PARTOF the generic body, but of being PARTOF Fred. If a depiction is instantiated more than once, this method of property inheritance allows us to keep track of which instance nodes are related to each other; for instance, if we wished to create instance nodes for Bill and Bill's arm, we would create a new binder of D-HUMAN, and use it to link the new instance nodes to their generic nodes; this ensures that Bill's arm will be PARTOF Bill, but not PARTOF Fred.

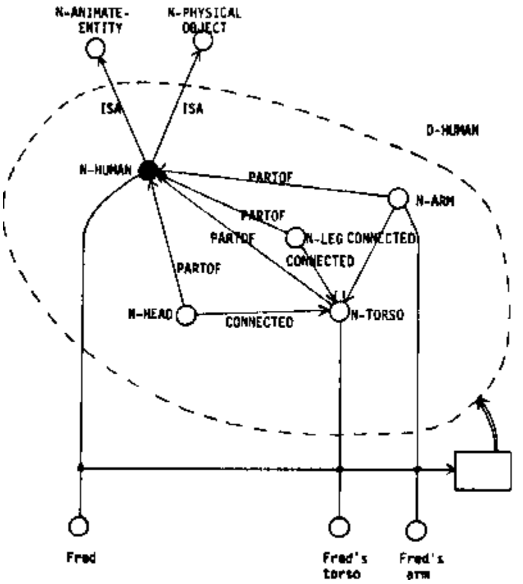

Figure ii   An instantiation of the depiction for a human

While binders instantiate depictions as a whole, it is not necessary, in any given binder, to provide instance nodes for every depicter; in particular, not all the nodes in D-HUMAN are instantiated by D-HUMAN#1. If, however, an instantiation of N-HEAD were added to D-HUMAM1, it would represent Fred's head, and both Fred and Fred's torso would inherit relations with it.

At this point, we can examine a first simple example of how depictions can help in finding disambiguating associations. Consider:

(1) Fred walked into the room; his arm was covered in bandages.

"Arm" is a word with many senses, but in this example it means a human arm, and furthermore Fred's arm. In the course of constructing a representation for the first sentence, CSAW would produce a binder of D-HUMAN to represent Fred; the correct representation for "his arm" is an instance node connected to N-ARM by the same binder. CSAW selects this instance node by using the fact that a node representing one sense of "arm" is in a depiction that has just been instantiated. A more general rule is:

(2) A node is associated with a depiction, if the node is in the depiction.

Since a context is defined in CSAW by a set of bound depictions, and since word-senses are represented by nodes, this rule also defines associations between word-senses and context. Rule (2) is only a first attempt at a definition of association, and will be much expanded below.

If the association of a node with a depiction is used to disambiguate an ambiguous word, then it is, of course, necessary to instantiate the node in a way that involves the binder which caused the depiction to be in the context. In the case of Rule (2), the involvement is straightforward; the node is simply instantiated in the same binder; this clearly provides the correct referent for "his arm" in Example (1). The expansion of Rule (2) will also involve an expansion of the corresponding instantiation procedure.

As mentioned above, association cannot always provide a unique referent, or even the correct sense; this point is illustrated by:

When I handed the hammer to the man, the (my, his) head fell off.

Nevertheless, Rule (2) and its elaborations, described below, have proved extremely useful when used in combination with CSAWs other disambiguating techniques, as described in [3].

III Generalizer Structure

One of the major advantages of semantic nets as a knowledge representation system is their ability to use hierarchies of ISA links to avoid duplication of information. If a particular fact is true of all the subsets of some particular set, then it need not be stored with each subset, but only with the superset. Thus Figure iii shows how the fact that horses, birds, humans, and dogs, all have legs can be represented by attaching that piece of information to the node for animal.

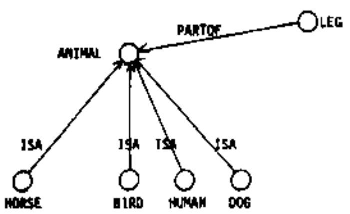

Figure iii A primitive property inheritance hierarchy

Problems arise, however, when the shared information has to be modified in some way. Suppose we want to represent how many legs each type of animal has. The obvious answer is to make a new node for the legs of each type of animal. An ISA link from each of these new nodes to the original leg node could indicate that they represent legs, but this is not enough; there must also be some way of telling which specialized leg node is associated with each animal subset. The obvious solution is a PARTOF link between the appropriate nodes, but in doing this we have duplicated precisely the piece of information we were trying to abstract. In the simple example given here, not much information is duplicated; in a more realistic example, there might be much more. Thus, besides a leg being PARTOF an animal, it is also used by the animal for locomotion. It would certainly be undesirable to store all this information as many times as there were subclasses of animal. We need a method of making the specialised leg nodes inherit all the relations postulated of the more general leg nodes. This difficulty is part of what Fahlman [2] has called the symbol-mapping problem.

One straightforward approach is to adopt the idea of binders. This requires that information about legs of animals be contained in an animal depiction, D-CREATURE, and that the depictions of the various animal subclasses contain binders of D-CREATURE as shown in Figure iv. The notation is essentially the same as before; note that the two binders of D-CREATURE are respectively inside D-HUMAN and D-DOG. To indicate the different role played by binders of this type, I will call them internal binders. Note also that there is no PARTOF relation between N-HUMAN and N-HUMAN-LEG. This relationship and all others between N-CREATURE and N-LEG are inherited through precisely the same mechanism as used for instance nodes.

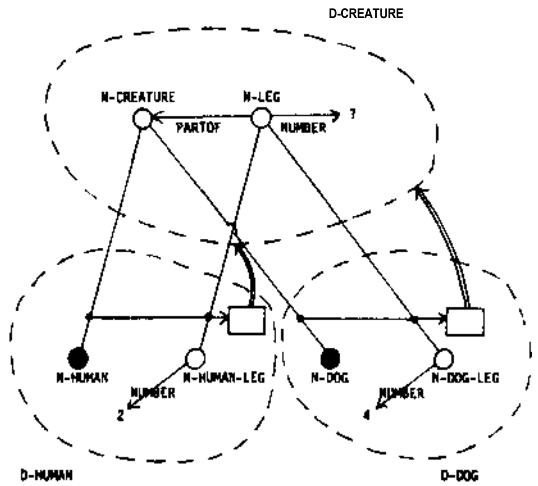

Figure iv A property inheritance hierarchy of depictions

Common information can be shared between the depictions of several different, but similar, objects in a much more efficient way: by using depictions as viewpoints. By making the links apparently attached to a node dependent on which depiction the node is viewed from, it is possible to use the same node to represent several different entities. Thus it is possible to use one fixed node to represent a human, a dog, or any other animal, and one other node to represent the leg of a human, or of a dog, or of any other creature.

This multiple use of nodes is implemented in CSAW in terms of a generalizer hierarchy of depictions. A generalizer hierarchy is a tree of depictions in which the parent depictions are said to be generalizes of their children. When one depiction is a generalizer of another, the offspring depiction automatically inherits all the links and nodes of the parent depiction. If any links or nodes are inappropriate for the offspring depiction, they can and must be specifically excluded. In addition, the offspring depiction can have extra nodes and links of its own, which are invisible from the parent depiction, even if they are links which involve nodes present in the parent depiction. The mechanism is thus similar to the one popularised by CONNIVER [8]; the difference lies in its intended purpose: in CONNIVER the trees of data environments were meant to represent alternative interpretations of some situation; in this system, depictions in a generalizer tree describe related but different objects. Depictions on the same branch of a tree will generally be more or less specific descriptions of objects that could be the same (animals and dogs): the nearer the root of the tree, the less specific the entity; while depictions on different branches of the tree describe entities which are different, and could not be the same, but are more or less related (humans and dogs).

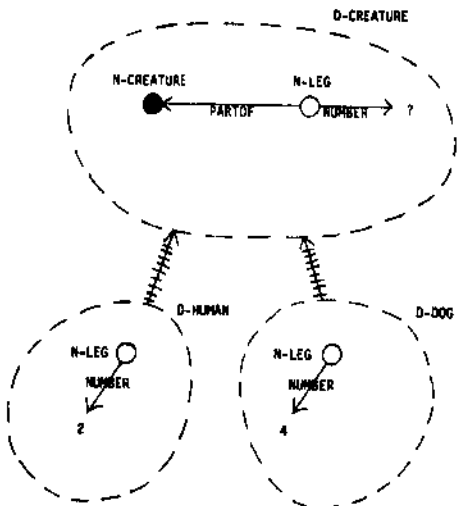

Figure v A generalizer hierarchy of depictions

To make this refinement clearer, let us consider how Figure iv would be transformed by it into Figure v. The crossed lines in Figure v indicate that D-CREATURE is a generalizer of D-HUMAN and D-DOG. Note that the PARTOF link in D-CREATURE is not written down again in D-HUMAN or D-DOG, since it is automatically inherited; this is also the case with N-CREATURE. N-LEG is also automatically inherited, but it is written down again in both D-HUMAN and D-DOG since it is involved in new links in both depictions. Note, however, that N-LEG in D-HUMAN and D-DOG is the self-same node as in D-CREATURE; it appears three times in Figure v only because of typographical considerations. The one point that Figure v fails to indicate is that the NUMBER link in D-CREATURE is not present in D-HUMAN or D-DOG because there is a specific directive to that effect associated with both depictions.

In the sense that there is an obvious translation of representations of ISA relations between depictions using the single-role node technique as exemplified by Figure iv and those using the multi-role node technique as exemplified by Figure v, the two approaches are equivalent. What then is the point in introducing the extra complexity of the latter approach? There are three reasons, given below in order of increasing importance. First, less space is used, since the number of nodes needed is considerably reduced, and there is no need to store pairings between nodes as in the single-role approach. Secondly, search time is reduced when trying to decide whether two nodes are connected by a given relation in a given depiction. In the former approach, besides looking in the depiction specified, it is necessary to make the enquiry recursively by translating it into an enquiry about the depiction which is bound in the lower depiction, changing the nodes enquired about according to the node pairings of the binder. In the latter approach no translation is necessary; it is enough to make the same enquiry in successively senior depictions of the generalizer hierarchy.

The third advantage does not concern general efficiency, but instead is specific to natural language applications and, in particular, the representation of word senses. In the multi-role technique, the sense of "leg" as leg of an animal corresponds to exactly one node, whereas in the single-role approach, it corresponds to many nodes: one for each type of animal represented. Consider the effect of this on the processing of:

> The dog was lying in the grass; one leg was bent in an odd direction.

According to Rule (2), the way to find possible referents for "one leg" in the second clause is to look in recently instantiated depictions for a generic node which can represent that sense. If we use the single-role approach, we have two options. First, we could store under the dictionary entry for "leg" all the leg nodes in all the depictions of animals, and then, when a referent was sought in this way, compare all these nodes with nodes in recently instantiated frames. This would certainly find the right answer, but the number of comparisons that would have to be made seems absurdly large. Alternatively, we could store in the dictionary entry only the node for "leg" that appears in D-CREATURE. The search for potential referents would then require that for each node in each recently instantiated depiction, we run up all the binder chains to see if they ended in the representative leg node. This is probably better, but still requires a lot of work. If, however, the multi-role node approach is used there is virtually no work to do. The appropriate sense of "leg" would be represented by N-LEG, and N-LEG appears in the depiction of dog, just as it appears in the depiction of *every* animal with legs.

One small complication arises because, with the multi-role approach, senses must be specified by a depiction as well as a node, The depiction is necessary because without it there would be no way to tell which links were attached to the node, and therefore no way to tell what the node represented. Such a combination of a node and a depiction to view it from is called a VNODE (viewed node); the VNODE of N-CREATURE viewed from D-DOG is written $<N-CREATURE,D-DOG.

In terms of finding associations, this means that after finding that the node part of a VNODE is in the depiction of a context binder, it is necessary to check that that depiction is compatible with the depiction of the VNODE. They are compatible if one is a direct or indirect generalizer of the other. Consider, for example:

Fred was in the room. His leg was covered in bandages.

After the processing of the first sentence, the context will contain a binder of D-HUMAN with an instance node for N-CREATURE to represent Fred. The dictionary entry for the appropriate sense of "leg" is $<N-LEG,D-CREATURE; because N-LEG is in D-HUMAN, and D-CREATURE is a generalizer of D-HUMAN, there is an association between this VNODE and D-HUMAN. The instance node which results from using this association is an instantiation of N-LEG in the binder of D-HUMAN constructed for the first sentence. The generalizer relationship is the other way round in:

I called the dog; the poodle refused to come.

This example seems slightly forced, and, in fact, examples in which the more general depiction comes after the less general one are much more common than vice-versa.

We can sum up the extension of Rule (2) to deal with VNODES as:

(3) A VNODE is associated with a depiction if its node is in the depiction, and its depiction is either a generalizer of or is generalized by the given depiction. In the first case, the node should be instantiated in the binder of the given depiction. In the second case, the binder should be converted into a binder of the depiction of the VNODE, and the node instantiated in the converted depiction.

This procedure is significantly less work than the procedure described above for use with the single-role approach, because the check for the node part of the VNODE being part of the context description acts as a filter on which generalizer hierarchies need to be searched. In the single-role approach any node in any depiction might potentially be connected to a given node through a chain of internal binders, the only way to find out is to investigate all the nodes connected in this way. In the multi-role approach, only those nodes which are the same as the node of a given VNODE need be considered for the subsequent generalizer check.

IV Multiple Subparts and SQN Structure

Next to the subset-superset relations expressed in generalizer structure, part-whole relations are the most important structural component of CSAW's data base. As explained below, part-whole relations, both physical and logical (a hand is part of a card game), can generate a new type of relation between depictions, which is independent of and orthogonal to the generalizer relation. It is best to introduce CSAW's treatment of part-whole relations independently of generalizer structure, and so for the moment, senses will again correspond to single nodes and nodes will again have single roles. The treatment does, however, require the use of depictions as viewpoints; so that the links attached to a node will still depend on which depiction it is viewed from. Instead of using this trick to let a single node represent several different but related entities, it will be used to allow different pieces of information about a single entity to be accessible independently. Figure vi shows a depiction D-ARM which contains the node N-ARM which also appeared in D-HUMAN in Figure i. In fact, N-ARMis the depictee of D-ARM, whereas it was a depicter in D-HUMAN. Note that the links:

(PARTOF N-ARM N-HUMAN)

(CONNECTED N-ARM N-TORSO)

that appeared in Figure i are not shown in Figure vi since they are not visible when N-ARM is viewed from D-ARM. However, a number of other links, which are similarly invisible in Figure i, are shown in Figure vi. The links shown in Figure vi are the links present in D-ARM.

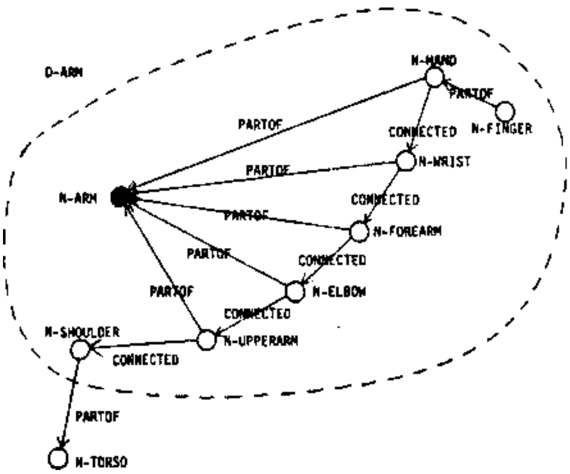

Figure vi An SQN depiction of an arm

Why make D-ARM a different depiction from D-HUMAN? Why should all the information about N-ARM not be put in D-HUMAN? There are two reasons. The first concerns levels of detail; if all the information about N-ARM is put into D-HUMAN, there seems no reason to stop there; all the information about bodies from eyebrows to toenails would be put in D-HUMAN, and in any given context most of that information would be irrelevant. Since it is clear from Rule (2) that instantiation of a depiction corresponds to an activation of all the information it contains, an instantiation of D-HUMAN, thus enlarged, would merely provide a confusing mass of irrelevant information. If instead such information is distributed over a collection of depictions chosen according to the "natural" division of a body into its various subparts, then only that information about bodies which is strictly relevant to a given context need be activated in that context.

The second reason is much more clear-cut, and concerns the representation of multiple parts. A normal body has two arms, but Figure i contains only one node, N-ARM, to represent them both; this paradox is resolved by allowing N-ARM to play a dual role: besides representing a generic arm, it also represents the generic set of all the arms of a body. Note that this sort of dual-role is quite different from the multiple roles used in generalizer structure; in the latter, a node represents one of a number of similar, but different, entities, depending on which depiction it is viewed from; in the former, a node represents both a set of similar entities which are all involved in a single depiction, and a typical member of that set. Such a node will play one role or the other according to which link it is used in. All the links involving N-ARM in Figure i use it in its typical member role. D-HUMAN should, however, be supplemented by other links involving N-ARM in its role as a set, in particular (NUMBER N-ARM 2). This link indicates the size of the set represented by N-ARM, and thus limits the number of instance nodes which may be paired with N-ARM in any given binder. Generic nodes without NUMBER links may be instantiated only once in a given binder (and the depictee will always be in this category), but those generic nodes with NUMBER links may be instantiated as many times in the same binder as their link indicates.

This method has several advantages over the alternative of using a separate generic node for each generic occurrence of a multiple subpart. If the latter method is used then:

a. If there are a large number of such occurrences there will be much wasteful repetition (legs of a centipede).
b. Since there is no single node which represents the subpart, a natural language dictionary entry for "arm" would have to treat the two generic arm nodes as two different senses of "arm". This problem could, however, be solved by introducing a further generic node representing the set of arms of a body.
c. The representation of "an arm" would require an arbitrary and unnecessary choice between left arm and right arm.
d. In the case of subparts which can vary in number, the maximum number would have to be represented. This could lead to the same problem as in a. for potentially large numbers, and even worse problems when there is no upper bound (the legs of a table).

Since there may be different generic information about them - left and right arms are indeed different - it is sometimes necessary to be able to distinguish between multiple subparts. This is accomplished through the use of <u>distinguishes</u>, a set of labels which span the set of subparts, in the sense that each generic subpart can be uniquely identified by a distinguishes or a collection of distinguishes. In the case of N-ARM, there are just two such labels, L-LEFT and L-RIGHT, corresponding to left and right. N-ARM distinguished by L-LEFT is written $!N-ARM(L-LEFT). Generic information involving only one or a subset of the subparts is handled by allowing links which apply to the node only when it is labelled by one of these distinguishes. An instance node is made specific by attaching the appropriate distinguisher to its pairing with its generic node in the appropriate binder. Of course, it is not necessary to distinguish if there is no basis for distinction (see c. above), and thus "an arm" in:

The man raised an arm.

could be represented by an undistinguished instantiation of N-ARM.

While there are compelling advantages to dividing the representation of a complex entity such as a body among several depictions, the fragmentation of such intimately related information results in certain complications. To deal with these complications, the relationships between the several depictions representing a complex entity are made explicit by incorporating them into an <u>SQN structure</u> (for sine qua non structure). The motivating idea is that if the depictee of a depiction is a subpart of, or in some other way implies the existence of, another entity, then that depiction is essentially incomplete. Its completion requires the depiction of that other entity in which its own depictee is presumably a depicter. Thus D-ARM does not make complete sense without D-HUMAN; we shall say that D-HUMAN is an SQN <u>depiction</u> of D-ARM. An SQN structure is a tree of depictions in which the parent depictions are SQN depictions of their children. All the depictions involved in representing a complex entity are typically tied together into an SQN structure. Clearly, SQN and generalizer structures are orthogonal in the sense that SQN structure is independent of and complementary to generalizer structure; the two types of structure are, however, similar in that they both create numerous separate trees, each of which groups a number of related depictions together.

SQN structure has important implications for depiction instantiation; whenever a depiction is instantiated, there must be a corresponding instantiation of any parent depiction it might have in an SQN hierarchy. Furthermore, one or more of the nodes in common between the two depictions can be designated as <u>SQNODES</u>; such nodes are instantiated by the same instance nodes in both binders. Thus an instantiation of D-ARM requires the instantiation of D-HUMAN; furthermore, N-SHOULDER and N-ARM, which are the SQNODES of

D-ARM, must be paired with the same instance nodes in both binders. Obviously, not all depictions have superiors in an SQN hierarchy; in particular, the depictions of stand-alone entities such as bodies do not. Such depictions will, in general, be the roots of SQN hierarchies. Thus D-HUMAN will be the parent of D-ARM, D-HEAD, D-LEG, etc., and D-ARM will in turn be the parent of D-FINGER.

The existence of SQN structure clearly requires a revision of our association rule (2). That subparts can be buried several layers deep in an SQN tree suggests the following replacement:

(4) A node is associated with a given depiction if it is either in the given depiction, or in a depiction which has the given depiction as a direct or indirect ancestor in an SQN hierarchy.

This rule would provide a disambiguating association for "nails" in:

Fred walked into the room; his nails were cracked and dirty.

in just the same way as Rule (2) provided one for Example (1). It was mentioned above that the binding of a depiction requires the creation of corresponding binders for any parent depictions in an SQN hierarchy; this requirement relieves Rule (4) from concern with offspring or sibling depictions in an SQN hierarchy.

v <u>Interaction</u> of <u>Generalizer and SQN Structures</u>

A generalizer hierarchy groups together depictions of similar, but different, entities. If such entities are the roots of SQN-hierarchies, their SQN-offspring will be similar, and so can be grouped into generalizer hierarchies of their own, which (approximately) parallel the top-level generalizer hierarchy. Figure vii, for instance, shows a small amount of knowledge about a small number of animals. There are two generalizer hierarchies (indicated by the barred lines): one for the top-level entities, the animals themselves, and the other for the limbs of the animals. Each animal depiction, except D-QUADRUPED, is the root of an SQN-hierarchy (indicated by the ordinary lines). In most cases, there is just one offspring node in these hierarchies, representing a typical leg of the animal in question. D-QUADRUPED does not have any SQN-offspring, since the description of a typical leg for a legged creature provided by D-LEG is quite adequate for the purposes of D-QUADRUPED. D-PRIMATE and D-HUMAN, on the other hand, each have two SQN-offspring: one describing their forelimbs or arms, and the other describing their rear-limbs or legs. That D-DOG only has one SQN-offspring indicates a lack of information in this representation for distinguishing the front and back legs of a dog. These differences in SQN-offspring mean that the limb generalizer hierarchy does not exactly parallel the one for animals.

Before extending the association rule to deal with both SQN and generalizer structure, the concept of a <u>depiction chain</u> must be introduced.

A depiction chain is a sequence of one or more depictions in which successive depictions are always related in one of a limited number of ways called depiction <u>kin-types</u>. For the moment, there will be just four kin-types, indicating relations between successive depictions in a depiction chain as follows:

SQN — the second depiction is the SQN-parent of the first;
SQN-1 — the first depiction is the SQN-parent of the second;
GEN — the second depiction is a direct or indirect generalizer-ancestor of the first;
GEN-1 — the first depiction is a direct or indirect generalizer-ancestor of the second.

For instance, according to Figure vii, D-DOG and D-LEG are connected by the depiction chain (D-DOG D-LEGGED-CREATURE D-LEG) in which the kin-types are GEN and SQN-1. The new association rule is:

(5) A VNODE is associated with a depiction if there is a valid depiction chain between the given depiction and the depiction of the VNODE.

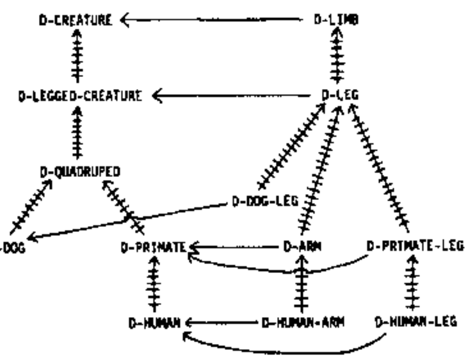

Figure vii Interacting generalizer and SQN hierarchies

If we assume for the moment that any depiction chain is valid, it is easy to see that the above rule subsumes Rule (3) and Rule (4), and so will work on the previous examples. The corresponding instantiation rule is:

(6) Given two consecutive depictions in a depiction chain and a binder of the first depiction, the production of an appropriately related binder for the second is accomplished according to kin-type as follows:

GEN — the given binder is used;
GEN-1 — the given binder is used after being changed into a binder of the second depiction;
SQN — a new binder is constructed for the second depiction in which the instance nodes for the SQNODES of the first depiction are the same as those in the given binder;
SON-1 a new binder is constructed for the

second depiction in which the instance nodes for the SQNODES of the second depiction are the same as those in the given binder.

Successive applications of Rule (6) will instantiate any depiction chain whose first member has a binder. The instantiation of an association, as defined by Rule (5), between a VNODE and a context depiction therefore consists of instantiating the depiction chain (using the binder of the context depiction to get started), and then instantiating the node of the VNODE in the final binder produced. To see how this rule works in practice, consider the association between the VNODE, $<N-F00T,D-LEG, for "foot" and the depiction D-DOG that would be required for the example:

I looked at the dog; its foot was swollen.

According to Rule (5) we need to find a depiction chain from D-DOG to D-LEG, the depiction of $<N-F00T,D-LEG. One obvious one in Figure vii is (D-DOG D-DOG-LEG D-LEG) with the kin-types SQN-1 and GEN. Instantiating this depiction chain according to Rule (6) would produce first a binder for D-DOG-LEG in which the SQNODES of D-DOG-LEG, say just N-LEG, were given the same instance nodes as in the given binder of D-DOG. Since the second kin-type is GEN, the binder of D-DOG-LEG just constructed would then be used unchanged as the binder for D-LEG. Finally, N-FOOT, the node of the VNODE, would be instantiated in this same binder to produce Figure viii, which is just what is required.

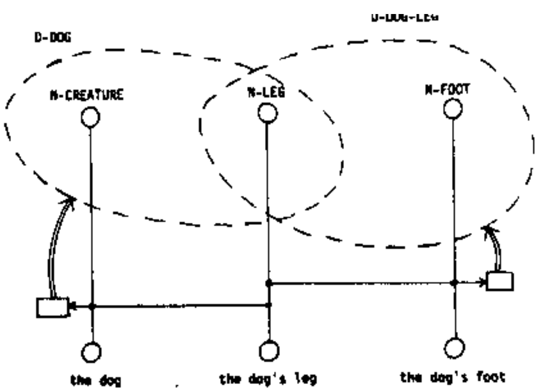

Figure viii Instantiation of depictions in interacting hierarchies

Of course there are other depiction chains from D-DOG to D-LEG, such as (D-DOG D-QUADRUPED D-HUMAN D-HUMAN-LEG D-LEG), whose instantiation would produce less desirable results. Such chains can be excluded by forbidding depiction chains with both GEN and GEN-1 kin-types or both SQN and SQN-1 kin-types. A more subtle type of error could be caused by the framechain (D-DOG D-LEGGED-CREATURE D-LEG). This would result in a diagram just like Figure viii except that D-LEG would be used instead of the more specific and appropriate D-DOG-LEG. This can be avoided by insisting that SQN and SQN-1 kin-types precede GEN and GEN-1 kin-types if possible. Trying to find an association between D-QUADRUPED and D-LEG shows that this is not always possible, but when it is not possible no error results.

VI Other Associations

Using just generalizer and SQN structure, one could produce a large number of descriptions of various stand-alone entities like animals or clocks or card games, etc., but one would have no way to relate these descriptions to each other. One way to increase the number of relationships represented is to increase the generality of the roots of the generalizer trees; thus one might incorporate the generalizer hierarchies for animals, plants, microbes, etc., into a single generalizer hierarchy for living things. However, such amalgamation must have a limit; otherwise, there would be only one generalizer hierarchy with the depiction D-ENTITY at its root, and only one node, N-ENTITY (or perhaps two, N-ENTITY and N-PART). Clearly, this is taking things too far.

Assuming that there must be separate generalizer trees for entities which do not display "significant" structural similarities, not all set-inclusion relations can be expressed by generalizer structure; the remainder are handled by internal binders as described in Section III. Besides representing set-inclusions for the depictee of a depiction, internal binders can also represent set-inclusions for the depicters and relations between the depicters. Figure ix shows internal binders performing all these functions; one internal binder in D-SHIP indicates that a ship is a vehicle, and, in addition, relates sailors to ships in the same way as drivers are related to vehicles in general; the other binder internal to D-SHIP indicates that the activity of sailing a ship is a form of disciplined activity involving the sailors. The diagram shows both N-BOSS and N-MINION paired with N-SAILOR; in fact, N-BOSS is paired with N-SAILOR distinguished to indicate officer, and N-MINION is paired with N-SAILOR distinguished to indicate non-commissioned sailor. The binder internal to D-VEHICLE shows that N-OPERATOR:VEHICLE is a human.

Such internal binders can provide important associations as can be seen from:

(7) When I visited the ship, the hands were out on deck.

Here "hands" is to be interpreted as the deck hands of the ship. If we use $<N-MINION,D-DISCIPLINED-ACTIVITY to represent the interpretations of "hand" as farm hand, deck hand, factory hand, etc., then the relation between N-SHIP and N-COMPLEX-ACTIVITY would provide exactly the right sort of association to make the correct disambiguation. Internal binders are, therefore, available as a valid kin-type, BIND, for use in the depiction chains of Rule (5). The association above can be expressed by the depiction chain (D-SHIP D-DISCIPLINED-ACTIVITY) in which the kin-type is BIND. The corresponding

instantiation rule is provided by the following fairly obvious supplement to Rule (6):

(8) BIND a new binder is constructed for the second depiction in which the nodes paired by the internal binder have the same instance nodes as the nodes they are paired with in the first depiction.

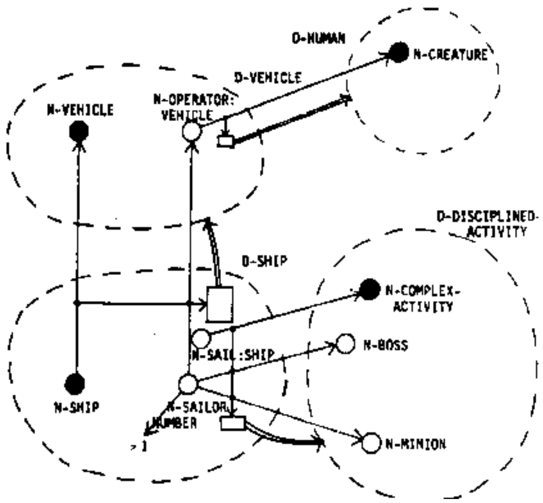

Figure ix Depictions with internal binders

Some restrictions must, however, be placed on the use of the kin-type BIND. Consider, for instance, the depiction chain (D-SHIP D-VEHICLE D-HUMAN D-HUMAN-ARM D-ARM) with kin-types BIND, BIND, SQN-1, GEN. This chain provides an association for $<N-HAND,D-ARM with D-SHIP, which could be expressed as: a hand is part of the one or more human beings who act as drivers for the sort of vehicle known as a ship. This association is too tenuous to be useful for the purposes of disambiguation, and in particular is not useful in Example (7). There is not enough space to discuss this problem fully here, and I will just state the solution I have adopted (see [3] for a justification). If a depiction chain contains a BIND kin-type corresponding to an internal binder with only one pairing, the chain is invalid unless the instance node for the nodes in the pairing represents something which has been explicitly mentioned in the text. Thus, the depiction chain mentioned above is invalid for Example (7), because the BIND kin-type between D-VEHICLE and D-HUMAN corresponds to an internal binder with only one pairing, and the nodes, N-OPERATOR:VEHICLE and N-HUMAN, in this pairing do not represent anything that appears explicitly in the text.

VII Conclusion

The system of data representation presented above was specifically designed for a natural language processing system which concentrates on finding the appropriate senses of ambiguous words in context. The associative structure provided by the system greatly facilitates the search for word senses that are associatively related to the preceding and surrounding text. In many cases, it can be used to provide referents for the ambiguous (or unambiguous) words which are found to be associated with the representation of the preceding text. The system is also interesting from a more general representational point of view; in particular, it embodies novel techniques for the representation of multiple subparts, and of objects that are almost, but not quite, the same.

The representation system has also been used for simple actions and the selectional restrictions they place on their participants. In addition it can help in making disambiguations based on the more specific associations demanded by simple actions and possessives. All this work is described in detail in [3].

References

[1] Bobrow, D.G. and Winograd, T., An overview of KRL, a Knowledge Representation Language, Xerox PARC, 1976.

[2] Fahlman, S.E., A System for Representing and Using Real-World Knowledge, AI Memo 331, Artificial Intelligence Lab., M.I.T., Cambridge, MA, 1975.

[3] Hayes, P.J., Some Association-Based Techniques for Lexical Disambiguation by Machine, doctoral dissertation, Ecole Polytechnique Federale de Lausanne, forthcoming.

[4] Hendrix, G., Expanding the Utility of Semantic Networks through Partitioning, Proc. 4th Int. Joint Conf. on Art. Int., Tbilisi, Georgia,"pp. TI5-T717T575.

[5] Minsky, M., A Framework for Representing Knowledge, in Winston, P. (Ed.), The Psychology of Computer Vision, McGraw Hill, 1975.

[6] Quillian, M.R., Semantic Memory, in Minsky, M. (Ed.), Semantic Information Processing, M.I.T. Press, Cambridge, MA, pp. 216-270, 1968.

[7] Simmons, R.F., Semantic Networks: Their Computation and Use for Understanding English Sentences, in Schank, R.C. and Colby, K.M. (Eds.), Computer Models of Thought and Language, W.H. Freeman, San Francisco, 1973.

[8] Sussman, G.J. and McDermott, D.V., From PLANNER to CONNIVER - A Genetic Approach, Proc. Fall Joint Computer Conf., AFIPS Press", MontvaTeTNJ, pp. 1171-1179, 1972.