To appear in Proceedings of the AAAI-94.

Unclear Distinctions lead to Unnecessary Shortcomings: Examining the rule vs fact, role vs filler, and type vs predicate distinctions from a connectionist representation and reasoning perspective

Venkat Ajjanagadde
Wilhelm-Schickard Institute, Universitaet Tuebingen
Sand 13, D-72076 Tuebingen, Germany
venkat@occam.informatik.uni-tuebingen.de

Abstract

This paper deals with three distinctions pertaining to knowledge representation, namely, the rules vs facts distinction, roles vs fillers distinction, and predicates vs types distinction. Though these distinctions may indeed have some intuitive appeal, the exact natures of these distinctions are not entirely clear. This paper discusses some of the problems that arise when one accords these distinctions a prominent status in a connectionist system by choosing the representational structures so as to reflect these distinctions. The example we will look at in this paper is the connectionist reasoning system developed by Ajjanagadde & Shastri(Ajjanagadde & Shastri 1991; Shastri & Ajjanagadde 1993). Their[1] system performs an interesting class of inferences using activation synchrony to represent dynamic bindings. The rule/fact, role/filler, type/predicate distinctions figure predominantly in the way knowledge is encoded in their system. We will discuss some significant shortcomings this leads to. Then, we will propose a much more uniform scheme for representing knowledge. The resulting system enjoys some significant advantages over Ajjanagadde & Shastri's system, while retaining the idea of using synchrony to represent bindings.

Introduction

Given a particular piece of knowledge, can one unambiguously decide whether it is a *rule* or a *fact*? Are there entities which always act as *roles* and never as *fillers*? Are there entities which always act as *fillers* and never as *roles*? What is a type and what is a general predicate?

In spite of the fact that the rule/fact, role/filler, type/predicate distinctions get mentioned not too infrequently in general AI parlance, an attempt to clearly state the distinctions faces difficulties (Some of the difficulties will be listed in the following section). This paper illustrates that taking these rather unclear distinctions and according them prominent representational status in a connectionist network may not be a desirable thing to do. Specifically, the example we consider here is the connectionist reasoning system(Ajjanagadde & Shastri 1991; Shastri & Ajjanagadde 1993) developed by Ajjanagadde & Shastri (Henceforth A & S). Their system performs an interesting class of inferences extremely fast. A major idea underlying their approach is the use of activation synchrony to represent dynamic bindings. We consider the idea of using synchrony to represent bindings to be indeed efficient, elegant, and as discussed in (Ajjanagadde & Shastri 1991; Shastri & Ajjanagadde 1993), neurologically plausible. However, the system of A & S has some shortcomings. These shortcomings are due to the representational methodologies A & S have chosen and are not due to the use of synchrony itself. The major reason for the shortcomings of their representational schemes can be diagnosed to be the prominence A & S have accorded to the distinctions of rules & facts, roles & fillers, types & predicates. The representational structures in their system directly reflect these distinctions. For example, Fig. 1 shows how A & S encode the following knowledge base:

give(x,y,z) $\Rightarrow$ *own(y,z)* ; *buy(x,y)* $\Rightarrow$ *own(x,y)*; *own(x,y)* $\Rightarrow$ *can-sell(x.y)* ; *give(john,mary,book1)*; *buy(mike,house3)*

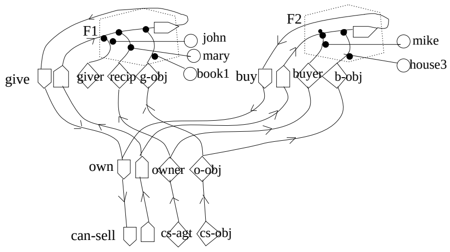

Fig. 1 An example network of A & S.

Fig.2 illustrates how A & S encode the following knowledge by interfacing the rule-based reasoner with a type hierarchy:

prey-on(x,y) $\Rightarrow$ *scared-of(y,x)* ; *prey-on(cat,bird)*; *isa(cat1,cat)*; *isa(cat2,cat)*; *isa(bird1,bird)*; *isa(bird2,bird)*; *isa(cat,animal)*; *isa(bird,animal)*.

[1] This paper was written in third person for double-blind reviewing.

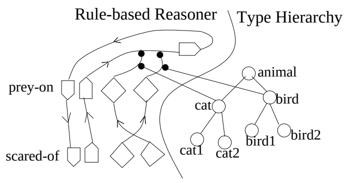

Fig. 2 Encoding the type hierarchy in A & S' system.

Now, note how the representational structures in A & S's system directly reflect the rule/fact, role/filler, and type/predicate distinctions. For example, note that "facts" are encoded in a way very different from the rules (e.g., look at the encoding of the fact *give(john,mary,book1)* (shown enclosed by the box F1 in Fig. 1) with the encoding of the rule *give(x,y,z)* $\Rightarrow$ *own(y,z)*). Similarly, A & S treat *role nodes* (e.g., in Fig. 1, *giver, recipient, give-obj, owner, own-obj,...*) in a fashion different from *filler nodes* (e.g., in Fig. 1, *john, mary, book1, house3,...*). The type/predicate distinction manifests in A & S's system as two different modules (shown separated by a curved line in Fig. 2).

Having recalled that the representational structures in A & S's system directly mirror the rule/fact, role/filler and type/predicate distinctions, let us discuss the appropriateness of according these distinctions such a prominent status.

A Closer Look at the Distinctions

Rules and Facts

What is a rule and what is a fact? Given a piece of information, can we clearly decide whether it is a fact or a rule? An intuitive response might be to say that rules correspond to general knowledge and facts correspond to specific knowledge ((Shastri & Ajjanagadde 1993), p. 418). Now, let us try to make that intuition a little more precise. A measure of the generality/specificity might be the number of individuals to which a piece of knowledge pertains to. So, if a piece of knowledge applies to a large number of individuals we may call it a rule and if the knowledge is about only particular individuals, we may call it a fact.

Now, let us imagine ourselves as having been given the task of representing a knowledge base as a connectionist network. For each piece of knowledge in the knowledge base, we have to decide whether it is a fact or a rule and choose the corresponding encoding scheme in A & S's system. Where do we draw the boundary between rules and facts? How general (specific) a piece of information has to be in order to be classified as a rule (fact)? We think that there is no such clear boundary and various pieces of knowledge fall in a continuous spectrum of generality rather than in two distinct bins. On the one hand, note that there are very few statements that apply to *all* individuals. So, for example, though A & S would consider the information that "When someone hits another, the 'hittee' gets hurt" to be a rule(Ajjanagadde & Shastri 1989), this statement does not really apply to *all* objects. For example, the 'hittee' has to be a sentient being to get hurt. On the other hand, though the knowledge "John loves Mary" may appear qualified to be called a fact (because it is about two particular human beings), that knowledge does hold about many particular instances of John and Mary: "John while wearing red shirt", "John while sitting in the pub", "John while having dinner" etc. still loves "Mary in blue skirt", "Mary while sitting in the pub". Hence what we refer to as "John" and "Mary" in the knowledge "John loves mary" correspond to sets of specific instances of John and Mary. Thus classifying an available piece of information as a rule or as a fact is not a clear-cut task. The unclear distinction between rules and facts gets blurred further in A & S's system with the interfacing of a type hierarchy with the rule-based reasoner(Shastri & Ajjanagadde 1993; Mani & Shastri 1991). They represent the knowledge "Cats prey on birds" as the fact *prey-on(cats,birds)*, which is a piece of knowledge about whole classes of cats and birds.

In summary, a representational scheme that forces us to divide the rather continuous spectrum of generality into two discrete bins does not seem appropriate.

Our proposal is to represent all the knowledge in the form of rules of the kind:

$$P_1(...) \wedge P_2(...)... \wedge P_n(...) \Rightarrow Q(...)$$

Now, representing more (less) specific knowledge is just a matter of having more (less) conjuncts on the antecedent of the rule.

With this choice of representation the knowledge that "John loves Mary" will be represented as the rule: *john(x)* $\wedge$ *mary(y)* $\Rightarrow$ *love(x,y)*.

In addition to the problem of conceptual clarity, the rule/fact distinction made in A & S's system leads to two other main shortcomings. The first one pertains to the ease of learning and the second pertains to reasoning power.

Learning: Learning involves generalizing from our specific experiences. One has to be able to learn "rules" from "facts". When significantly different kinds of network representations are used to represent specific experiences and more general knowledge, learning gets harder. Starting from one kind of representation and arriving at a significantly different kind of representation is a difficult thing to achieve with only local readjustments that are normally made use of in connectionist learning. When we use a uniform representation, generalizing (specializing) corresponds to dropping (adding) some conjuncts in the rule and can be achieved by simply weakening (strengthening) the relevant links.

Reasoning Power: Note that in the network of Fig. 1, there is an asymmetry in the flow of bindings between rules and facts. Binding information can flow from the rules into facts, but not vice versa. Specifically, note that there are no distinct connections from the filler nodes involved in a fact to the corresponding role nodes. For example, consider the encoding of the fact *give(john,mary,book1)*. There are no (distinct) connections from the filler nodes *john, mary, book1* to the role nodes *giver, recipient, give-obj*. That means that facts cannot induce bindings in the argument nodes. Note that, in general, when we are unifying two expressions, we would want bindings to flow in a bidirectional fashion. So, for example, when unifying $P(a,x)$ and $P(y,b)$, we want x to be bound to b and y to be bound to a. But, this does not happen in A & S's system. A specific example where such bidirectional transfers of bindings are necessary corresponds to "answer extraction". Suppose we ask the query $?\exists x$ *can-sell(mary,x)* with respect to the network of Fig. 1. We would expect x to get bound to *book*1 (That way, we know not only that "Mary can sell something" but also what can Mary sell.). This binding has to come from the fact *give(john,mary,book1)*. But, since in the network of Fig. 1, facts cannot induce bindings, one can only prove that $\exists x$ *can-sell(mary,x)* is true, but, cannot get the answer *can-sell(mary,book1)*. This is the reason why A & S add extra circuitry (not shown in Fig. 1) and a two stage process to get back such answers (see section 4.7 in (Shastri & Ajjanagadde 1993)). When we adopt a uniform scheme for representing rules and facts, the asymmetry between rules & facts goes away, and bidirectional transfer of bindings is naturally obtained. "Answer extraction" happens to be a special case benefit (without having to use additional circuitry for that purpose) of such bidirectional transfer of bindings.

Rule-based reasoner and type hierarchy

Let us consider representing the information "Cats prey on birds". One can represent this knowledge as the rule "$\forall x \forall y$ *cat(x)* $\wedge$ *bird(y)* $\Rightarrow$ *prey-on(x,y)*. But, A & S choose not to do this ((Shastri & Ajjanagadde 1993), p. 435). Instead, they introduce filler nodes corresponding to "cats" and "birds". These nodes behave similar to the nodes corresponding to individuals such as *john*, *mary* etc. Hence, similar to the way one represents facts about individuals (e.g., *love(john,mary)*), one can represent the knowledge "Cats prey on birds" as the fact *prey-on(cat,bird)* (Fig. 2). Why do A & S go for the extra trouble of interfacing the rule-based reasoner with a type hierarchy, when they could represent the same information as just another rule? The reason they give is "... The rule-based reasoner ... cannot answer queries such as prey-on(cat,bird)"((Shastri & Ajjanagadde 1993), p. 435). That is, one could not ask their rule-based reasoner queries about classes of individuals. But, as we will discuss in the next section, with the representation and reasoning scheme we are proposing, the rule-based reasoner itself can answer queries such as "Do cats prey on birds?", "Are birds scared of cats?". In fact, queries of this kind happen to be just special cases of the queries that can be handled by the system we are proposing.

Having said that one does not need to go for a type-hierarchy interface to be able to ask the kind of queries A & S mention, we would like to go one step further and note that the idea of type-hierarchy interface is not just redundant, but, a handicap too. There are four reasons why this is so. Firstly, as mentioned above, with the scheme we are proposing one can have more general queries than the ones A & S consider. With A & S's approach, these general queries cannot be handled.

Secondly, with the idea of type hierarchy interface, only *isa* knowledge can be used to make inferences about types. But, we do need an ability to infer types based on relational knowledge. For example, when we see particular spatial relationships between three blocks, we would like to be able to infer that the the arrangement formed by the three blocks is of type 'arch'. Due to the way relations (predicates) and types are separated into two different modules in A & S's system, the knowledge of relations is not used to make inferences about types.

Thirdly, there are conceptual problems with the idea of type-hierarchy interface. What is the distinction between types and predicates? If they are not completely distinct and there is a semantic overlap, how do we represent that overlap?

Fourthly, the comment made in the previous subsection about the advantage of having uniform representations from the point of view of learning, apply here as well.

Roles and Fillers

The role vs filler distinction in A & S's system corresponds to the term vs predicate distinction in first-order logic. First-order logic represents the world as a set of objects and allows one to make assertions about those objects. The terms of first-order logic correspond to the objects and predicates are used to make assertions about those objects. In A & S's rule–based reasoner (without the type hierarchy interface), "filler nodes" correspond to objects in the world and "role nodes" are used to make assertions about those objects. Thus, the roles vs fillers distinction in A & S's system is indeed clear and well motivated to the extent the predicate vs term distinction in first-order logic is. Having said that the distinction between role nodes and filler nodes is indeed clear in that way , we would like to make two observations.

Firstly, it is relevant to note that sometimes, one wants to make assertions about not just objects in the world but also about concepts. It is not so straight forward in first-order logic to do the latter. Two of the classic examples illustrating this problem are the "telephone number problem" and the "morning star problem"(McCarthy 1979). Following A & S's approach, in order to represent assertions such as "John knows Mike's telephone number", one will have to have a filler node for the intention of "Mike's telephone number". When one does that (i.e., relaxes the condition that filler nodes correspond to objects in the world), the distinction between role nodes and filler nodes gets blurred[2].

Our second observation concerns plausibility under resource constraints. As mentioned above, in A & S's system, filler nodes denote individuals in the world. But, it is obvious that there cannot be a distinct node corresponding to every object we ever reason about. The number of such individual objects is extremely large (virtually infinite). It is infeasible to say that there is a distinct node in the network corresponding to every such individual. A & S indeed take this fact into consideration. Avoiding the need to allocate a unique node corresponding to every object is precisely the reason they use activation synchrony to represent dynamic bindings. In A & S's system, a phase corresponds to an object participating in a particular reasoning episode. All the nodes active in that phase *together* represent that individual. Synchrony is a means of representing the grouping of the nodes representing the features of an object. Thus, the idea of representing an object by a group of nodes instead of a single node is indeed present in A & S's system though they do not exactly describe it in this way. However, while A & S do not assume the existence of a distinct node corresponding to every object, they do assume the existence of a distinct node corresponding to *every object that is involved in a long-term fact.* Thus, in Fig.1, for representing the long-term facts *give(john,mary,book1)* and *buy(mike,house3)*, A & S assume the existence of distinct nodes corresponding to the objects in these facts, namely, *john, mary, book1, mike, house3.* A logical extension would be to abandon altogether the idea of having distinct nodes corresponding to individual objects (i.e., irrespective of whether or not there is a long-term fact about that individual). With this suggestion, the very notion of *filler nodes* (i.e., nodes corresponding individual objects) looses its utility.

An Alternative Scheme of Representation

In the scheme we are proposing, there are no assorted types of nodes such as role nodes, filler nodes, collectors, enablers etc. that A & S make use of. Instead, the nodes in the network are all *feature nodes.* Each node corresponds to a *basis feature.* An object is represented by a group of basis features. "Basis features" are unary features and are akin to what are usually referred to in connectionist literature as *microfeatures.* We are using a different terminology just because we suspect that some of what we mean by a "basis feature" may be in disagreement with what some people may mean by a "microfeature". For example, it is not necessary that there should not be a subsumption relation between one basis feature and another. For instance, *human* and *animal* could both be basis features. Even though it might be possible to represent some feature (say, *human*), by a combination of the already existing basis features (say, *animal, biped,...*), an agent may still have a node corresponding to that feature. The nodes in A & S's network happen to be a rather extreme special case wherein *john, mary, book1* etc. themselves happened to be basis features.

It is perhaps worth mentioning that this paper is not concerned with what the basis features are; instead, the focus is on how to represent knowledge as an interconnection of the nodes corresponding to basis features and how does reasoning take place in that network. Generally, in our examples, we will be taking the nodes in A & S's networks themselves to be basis features. This will help one to contrast and see how our representational scheme differs from that of A & S even if we hadn't brought in the idea of not having to have distinct nodes corresponding to individual objects. Thus, in Fig.3, *giver, recipient, owner, own-object, ...* etc. themselves have been chosen as basis features.

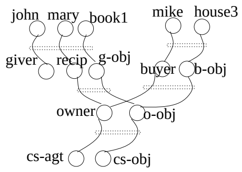

Fig. 3 Network as per the proposed scheme that corresponds to A & S' network of Fig. 1.

Having said what individual nodes in the network represent (i.e., basis features), let us now discuss the encoding of rules. Corresponding to every rule, there are a group of links, which we will refer to as a *link bundle.* In the figures, we denote *link bundles* by drawing a thin bar over the links forming the bundle. For example, the links between the nodes *buyer, buy-obj* and the nodes *owner, own-obj* form a *link bundle* representing the rule *buy(x,y)* $\Rightarrow$ *own(x,y).* The actual interconnection details corresponding to this rule are shown in Fig. 4(a). To avoid clutter, we will normally depict link bundles as shown in Fig. 4(b).

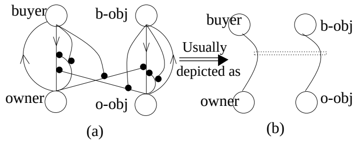

Fig. 4 Encoding *buy(x,y)* $\Rightarrow$ *own(x,y)*

In fact, that is all we need to say to describe the details of encoding in our network. Unlike in the system of A & S, we do not need to separately explain how rules are encoded, how facts are encoded, how the type hierarchy is encoded etc. All that is there

[2] For a discussion of some related issues, see (Wilensky 1986), where Wilensky examines some of the distinctions often made in frame-based systems and then, argues for a more uniform representational scheme.

in our network are just these: There are nodes corresponding to basis features and then there are rules (which are represented by link bundles). Facts and type hierarchy knowledge are not encoded in a different way; they get encoded as rules. For example the network of Fig. 3 includes the encoding of the fact *give(john,mary,book1)*. This fact is encoded as the rule: *john(x)* $\wedge$ *mary(y)* $\wedge$ *book1(z)* $\Rightarrow$ *give(x,y,z)* . This encoding assumes that *john*, *mary*, and *book*1 are basis features. But, that need not be the case; these individuals can be represented in terms of some other basis features. Suppose, for example, that they are described using the following features: *john*:*black-hair,round-face* ; *mary*:*blond,long-face*; *book*1:*book,thick,red.* With that representation of objects, the encoding of the fact *give(john,mary,book1)* is shown in Fig. 5.

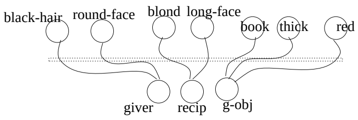

Fig. 5 Encoding *give(john,mary,book1)*.

The information represented in A & S's network of Fig. 2 is represented in our network as shown in Fig. 6. This network encodes the following rules: *prey-on(x,y)* $\Rightarrow$ *scared-of(y,x)* ; *cat(x)* $\wedge$ *bird(y)* $\Rightarrow$ *prey-on(x,y)*; *cat1(x)* $\Rightarrow$ *cat(x)*; *cat2(x)* $\Rightarrow$ *cat(x)*; *bird1(x)* $\Rightarrow$ *bird(x)*; *bird2(x)* $\Rightarrow$ *bird(x)*; *cat(x)* $\Rightarrow$ *animal(x)*; *bird(x)* $\Rightarrow$ *animal(x)*

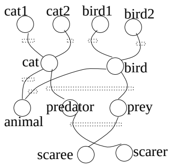

Fig. 6 Network as per the proposed scheme corresponding to the network of Fig. 2

Reasoning

Having discussed the encodings in the network, let us look at how reasoning takes place.

The general form of querying in our system is the following: One asks the network "If I now assert that P_1, P_2,...,P_k are true about a set of objects, then, can you prove Q using P_is and the knowledge already encoded in the network?". We will refer to P_is as (dynamic) assertions and Q to be the target proposition. Note that one has to somehow distinguish P_is from Q. This is because P_is are being asserted to be true while Q's truth value is what we want to find out. We represent this distinction by using high activation level for representing P_is and a low activation level for Q. If Q can indeed be proved to be true, then, its representation attains high activation level in time dependent on the length of the proof.

Now, let us consider a simple single step inference. Suppose we want to query the network of Fig. 3 to find out whether "buy(mike,house3)" is true. We pose this query as follows: *Suppose we assert that* $mike(obj_1)$ *and* $house3(obj_2)$ *are true, can you prove that* $buy(obj_1, obj_2)$ *is true?*[3] That is, the assertions in this case are: *mike*(obj_1) and *house3*(obj_2). The target proposition is *?buy*(obj_1, obj_2).

To pose this query, we clamp the activity patterns representing the assertions at a high level of activity and clamp the activity pattern corresponding to the target proposition at a low level of activity.

Suppose we associate the first and second phases with obj_1 and obj_2 respectively. In that case, clamping the assertions and target propositions involves doing the following: We make the nodes corresponding to *mike* and *house*3 to become active in the first and second phases of every cycle (respectively) at a high level of activity. We make the nodes corresponding to *buyer* and *buy-obj* to become active in the first and second phase of every cycle respectively, but at a low level of activity. If the target proposition (i.e., *buy*(obj_1, obj_2)) can indeed be proved to be true (which is the case in our example) we would expect the activity levels of *buyer* and *buy-obj* to become high after a while. Let us examine how this indeed happens.

Since the node *mike* is firing at a high level of activation (in phase 1), unless the flow of activity along the link A (Fig. 7) from *mike* to *buyer* is inhibited, *mike* will raise the activity level of *buyer*. But, there indeed are two inhibitory connections onto this link: link C from *buyer* and link D from *buy-obj*. In phase 1, the node *buyer* is active hence could potentially inhibit the flow along link A. But, note that in phase 1, *mike* is active as well and hence the activation flow along link E inhibits the flow along link C. As a net result, there will be no inhibition on link A in phase 1. Reasoning along similar lines, one can find that though *buy-obj* becomes active in second phase, it does not succeed in inhibiting link A. That means that there will not be inhibition on link A during any phase of the cycle [4]. Hence, the flow of activity from *mike* along link A takes place thereby raising the activity level of *buyer*.

Analogously, one can see that the flow of activity from *house*3 raises the activation level of *buy-obj*.

---

[3] We use obj_is to denote arbitrary objects. The situation is similar to starting a mathematical proof by a statement such as "Let x be an arbitrary integer". To make it sound even more analogous to our situation, consider a paraphrase of that statement, namely, "Let x be an arbitrary number having the feature of being an integer". Quite analogously, the assertion $mike(obj_1)$ for example corresponds to asserting "Let obj_1 be an arbitrary object having the feature of being *mike*".

[4] It is important to note that we assume that for activity flow to take place along a link, there should not be inhibition on that link during *any* phase of a cycle; we assume that an inhibitory effect lasts for the duration of a cycle.

As a result, the target proposition *buy*(obj_1, obj_2) indeed gets proved to be true.

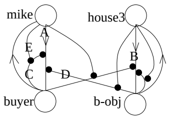

Fig. 7 Encoding of *mike(x)* $\wedge$ *house3(y)* $\Rightarrow$ *buy(x,y)*.

The essential thing to note about the inhibitory connections in a link bundle (see e.g., Fig. 7) is that they are designed to check the binding consistency between the antecedent and the consequent of the rule. When there is a binding mismatch, all the links from the feature nodes corresponding to the antecedent of the rule to the feature nodes corresponding to the consequent of the rule get inhibited.

Having seen how a single-step inference involving a rule takes place, it is easy to see how multi-step inferences take place in the network. Regrettably, due to space limitation, it is not possible to take the reader through some more examples. But, the information provided so far should be sufficient for a reader to check how the system works for other examples. One particular point that is to be remembered however is that we are assuming here that only one instance of a feature needs to be represented during a reasoning episode. The issue of representing multiple dynamic instances(Shastri & Ajjanagadde 1993) is quite orthogonal to the subject matter of this paper. Two particularly interesting examples to consider pertain to the bidirectional transfer of bindings and dealing with queries such as "Are birds scared of cats?". Consider asking the query *?*$\exists x$ *buy(mike,x)*. This corresponds to the case when the assertion is *mike*(obj_1) and the target proposition is $\exists x$*buy*(obj_1*,x*)[5]. It may be noted that if the object *house*3 happens to be in focus, i.e., if the node *house*3 happens to be active, say, in some phase i, then, the node *buy-obj* will also start firing in phase i representing the desired answer that the thing Mike bought is "*house3*".

Now consider the query "Are birds scared of cats?". In this case, the assertions correspond to *cat*(obj_1) and *bird*(obj_2). The target proposition is *?scared-of*(obj_2, obj_1). One may work through the network of Fig. 6 to see that this indeed produces the desired answer[6]. This illustrates that our scheme can answer queries about classes of objects without needing a type hierarchy interface. The reason for saying that our scheme can deal with even more general queries than A & S's system is the following: The queries A & S are able to deal with the type hierarchy interface correspond to the case when assertions involve only unary predicates such as, for example, *cat*(obj_1) and *bird*(obj_2). In addition to such assertions, in our system the assertions can also involve n-ary relations such as *P*(obj_1, obj_2).

[5] As in A & S's system we do not activate the nodes corresponding to the unbound arguments in the target proposition.

[6] Note that obj_1 and obj_2 are arbitrary objects and the only thing assumed to be known about them is that *cat*(obj_1)) and *bird*(obj_2). That is, obj_1 and obj_2 are arbitrarily chosen members of the classes of *cats* and *birds* respectively. Hence, if we can prove that *scared-of*(obj_2, obj_1), that means that birds in general are scared of cats.

Conclusion

As concluding remarks, let us summarize the intended contributions of this paper.

Firstly, the paper provides a critique of some of the representational structures employed in the connectionist reasoning system of Ajjanagadde & Shastri. While this critique of the specific system by A & S should be of interest by itself, it is our belief that the issues raised are of general interest and merit consideration in connectionist knowledge representation efforts in general.

Secondly, the paper proposed an alternative scheme for representing knowledge and presented a system that enjoys some significant advantages relative to the system of A & S in reasoning ability, conceptual clarity, ease of learning, representational efficiency and neurological plausibility.

Acknowledgments. This work was supported by DFG grant Schr 275/7-1. I would like to thank my colleague Seppo Keronen for numerous useful discussions spread over many lunches and afternoon teas.

References

Ajjanagadde, V. G., and Shastri, L. 1989. Efficient inference with multi-place predicates and variables in a connectionist system. In *Proceedings of the Conference of the Cognitive Science Society*, 396–403. Lawrence Erlbaum.

Ajjanagadde, V. G., and Shastri, L. 1991. Rules and variables in neural nets. *Neural Computation* 3:121–134.

Mani, D. R., and Shastri, L. 1991. Combining a connectionist type hierarchy with a connectionist rule-based reasoner. In *Proceedings of the Conference of the Cognitive Science Society*, 418–423.

McCarthy, J. 1979. First-order theories of individual concepts and propositions. In Hayes, J.; Michie, D.; and Mikulich, L., eds., *Machine Intelligence 9*. Halstead Press : New York. 129–147.

Shastri, L., and Ajjanagadde, V. G. 1993. From simple associations to systematic reasoning: A connectionist representation of rules, variables, and dynamic bindings using temporal synchrony. *Behavioral and Brain Sciences* 16:417–494.

Wilensky, R. 1986. Knowledge representation – a critique and a proposal. In Kolodner, J., and Riesbeck, C., eds., *Experience, Memory, and Reasoning.* Erlbaum: Hillsdale, NJ.