DISTINCTIONS AND CONFUSIONS: A CATALOGUE RAISONNE*

David J. Israel
Ronald J. Brachaan

Bolt Baranetc and Nevaan Inc.
Cambridge, HA 02238

ABSTRACT

It's been said many times that semantic nets are mere notational variants of "predicate calculus". But before we lay down our nets, and embrace Logic, we ought at least to be clear about what "predicate calculus" is. We advance some clarificatory points to this effect. Moreover, there seems to be at least one feature of some net/frame schemes that defy simple translation into standard formalisms: "prototypes". Yet this feature is not itself without difficulties, some of which we address here. In the end, we plead for an open mind and a search for formal-semantical alternatives to "classical logic".

I. INTRODUCTION

With "knowledge representation" becoming a hot topic in Artificial Intelligence in recent days, many a barroom discussion at an AI conference has turned to the relationship between logic and the "new" AI representation languages. Sitting across the table from one another we might find a hard-core semantic nets aficionado and a born-again logician, debating the relative merits of nodes and links, predicates and variables. The discussion might get heated; somehow, despite an inability to prove that his network scheme has greater expressive power than first-order logic, the net-worker "knows" that there is something more to his nets than is dreamed of in the "logician's" philosophy.

Who shall we believe in this tangled dialectic? We will contend here that it's not even clear what the competing positions are. We first take a brief look at the ancestry of semantic nets and see how they've been left a cloudy legacy. The greatest degree of clarity on the meaning of semantic nets seems to have come from those who view them as mere notational alternatives to more traditional logical notations, so we examine some of the positions adopted by these "logicians". Unfortunately, we find them to be a bit hazy as well. Lest our confusion over the "logicians'" position be taken for a defense of semantic nets, we ultimately turn back to the "network hackers" for some critical comments. In particular, we cast a sidelong glance at "prototypes", illustrating how their primacy in networks has perhaps done those formalisms a disservice. In the end, we throw our own hat into the ring, advocating, like the logicians, a strong formal-semantical approach. But we refuse to be forced into straitjacket of "classical logic". We conclude by offering some extremely brief suggestions of alternative ways of looking at representation. These are at best programmatic, and may not do much to counteract the overall negative tone of the paper, but we do feel that the field needs a critical overview before the brawl spreads to the whole bar.

*This research was supported in part by the Defense Advanced Research Projects Agency, and was monitored by ONR under Contract No. N00014-77-C-0378.

II. A BRIEF TENDENTIOUS HISTORY OF SEMANTIC NETS

Since "semantic networks" are the point of contention in our barroom debate, we start out by explicating their somewhat confusing state with a little history. A brief ramble through their ancestry will indicate some of the traits inherited from a succession of what are now apparently mixed marriages.*

First, of course, there was Quillian's work on "Semantic Memory". Quillian used a network structure to model human verbal memory. His intent was to get "humanlike use" by processing the network of associations with a spreading activation search. Many of the mainstays of current network representations were present in that first venture, including descriptions in the general-class-plus-modifier tradition.

Collins and Quillian followed up the original work with a series of psychological plausibility tests for network models, using reaction times to test how many "levels" intervened between stored facts. With a couple of years of work with strictly psychological motivation, then, semantic nets were off and running.

The next big move on network notations came from a natural language processing point of view. (Quillian had had language in mind from the very start, it should be noted.) In particular, in several new network systems, nodes came to be looked at as surrogates for verbs and nouns, with links standing for semantic "case" relations. For the most part, the nets were still intended to be psychologically plausible models, although some formal considerations were starting to creep in.

At more or less the same time as the popularization of networks for linguistic case frames, nodes were being used in other work to denote (physical) objects in the (blocks) world, and links to represent relations among those objects. This in fact represented a significant departure in the interpretation of the network

*This is of necessity a rough-and-tumble trip - see [4] for a more comprehensive survey, with references. Also, see [7], pp. 14-18 - especially the comments by Goldstein - for more about the confusing hand that History has dealt to semantic nets.

notation. Modes were now supposed to stand for structured object* in virtue of their own semantically significant structure. Moreover, the connection with verbal memory was being severed. While the nets were still strictly <u>notations</u> and not languages, in this use they began to resemble other* established formalisms - notably formal quantifloatlonal languages.

*Two devel*opments in the middle 1970's contributed the critical chapter in the semantic nets story. Minaky's "Frames* paper [16] directed tremendous attention to default-oriented reasoning with large-size chunks of information. In many network notations, nodes came to resemble "frames". The prominent feature of these notations was that the nodes stood principally for "prototypes" (e.g.. TYPICAL-ELEPHANT), and once again, psychological concerns were reasserted as the motivation for much of the memory organization and reasoning.

The other influential development was the appearance of Woods' "What's in a Link" paper [25], an attempt to examine the "semantics" of semantic nets. Woods showed that ofttimes net-workers had been vague, and their network "languages", semantics-less. But perhaps the paper's most significant contribution to the field was the subtle shift in perspective it induced - it in essence granted semantic networks full-fledged "representation language" status. The consequence of Woods' assumption (that semantic nets <u>should</u> have semantics) was that a great number of people took seriously the challenge of providing their network formalisms with the appropriate well-specified semantics. In particular, one line of development that followed almost immediately was a series of network notations modelled directly on the language of first-order logic (e.g., *[22]).* Other efforts took Woods seriously in other ways. providing us with an interesting diversity of Interpretations of the "semantics of semantic nets" (for example, of. KL-ONE, KRL, NETL, and PSN's).

III. <u>DRAWING THE BATTLE LINES</u>

The variegated background of semantic nets has left us in this state: most people now see them as "representation languages", despite the fact that they were Initially intended primarily as psychologically testable memory models - not as languages at all. For some, the expressive power of the nets is to be psychologically unconstrained and perchance universal; some see nets as peculiarly tied to verbal comprehension; others, though unmindful of this latter connection, remain faithful to the original psychological motivation, typically by taking prototypes as their central focus. And almost everyone now takes Woods' dictum - specify your semantics - as gospel. In fact, some zealous net-workers feel that the only way to answer that challenge is to base networks directly on well-understood formal languages. Some (usually non-net-workers) go even further, and claim that semantic nets are nothing other than notational variants of first-order languages.

Just for argument's sake, let's divide the field as it currently stands into four factions:

1* <u>logicians</u>: former net-workers who have now accepted Logic, as well as those who never were net-workers and have always known The One True Logic [sic].

2- <u>net-logicians</u>* those who claim their networks are simply notational variants of standard first-order languages, perhaps with special indexing considerations.

3. extended-net-lQgiclana: those whose nets are variant notations for ^ standard quantifloatlonal languages, but with allegedly minor extensions.

4. Mtrrarkora: tho»« whoa# ■**• **** not claimedto be variants of standard quantifloatlonal formalisms, or are explicitly claimed not to be. This group breaks down into two sub-factions:

  o pflYfihprnmt-wwrkara those whose networksare strictly psychologically motivated.

  o DQa-paYfiho^Mt-Wwrkers: those whose netsareoriented towards psyohologioally unconstrained universal expressive power.

As we hinted above, perhaps the clearest response to Woods' oonoern about the semantios of semantlo nets has come from the first three of these groups. In the following section we analyze in some depth the conoerns and assumptions of these "logical" types.

IV. *QML* <u>«MAWTf! METS AMD</u> »<u>piMT-QBD«B LOQIC</u>«

Imagine believing that "semantlo net notation. i.e., the graphical symbolism used by semantlo net theorists, is a variant of traditional logical notation" [23, p. 122]. Why then go to the trouble of being a semantlo net theorist?* Far and away. the most commonly cited reason has to do with access and retrieval features. Witness Shapiro, as cited in [23i p. 122]: "All the information about a given conceptual entity should be reachable from a common place." In a network, knowledge about a given entity (or a given kind thereof) is "directly attached" to the node for that entity (kind). Indeed, Sohubert, et al. enunciate a fundamental assumption that motivates the deployment of semantic nets: "(T)he knowledge required to perform an intellectual task generally lies in the semantlo vicinity of the concepts involved in the task." [23, P. 123].

This bundling feature is quite distinct, say such net-logiolans, from the networks considered purely as representational formalisms; that is, the aocess and retrieval features do not distinguish network formalisms from their "traditional" counterparts in respect of expressive power.•• Indeed, we are reminded to keep quite distinct in our minds the following aspects of a representational system:

1. the purely representational or expressive features of the formalism involved;

2. the deductive apparatus (calculus) defined over the formalism Tor more broadly, any set of syntactically-specified rules of transformation);

3. an algorithm implementing (2), which might involve

4. access and retrieval procedures defined over the formalism.

•We assus that by "traditional logical notation", the authors mean languages of the standard quantifloatlonal variety. This assumption seems wildly well warranted.

••Note that it is one thing, to treat a formalism as a mere notational variant of a standard language; it is quite another to treat it as a different language equivalent in expressive power to suoh a language.

[We will soon add another distinction: that between a language and a logic.] In sum, our logical friends aver that networks and "predicate calculus formalisms"* are equivalent on (1), but not necessarily on any of (2) through (4).

A. On Standard Quantificational Formalisms

Now that we've been told, at least with respect to representational issues, to boil everything down to (standard) quantificational formalisms, all would be peachy if only it were agreed on all sides what the sediment was. In particular, what does one embrace when one accepts a "standard" quantificational formalism? Darned if we know (see section IV.C). But it is pretty clear what you get when you buy a standard quantificational language: mostly, just some syntactic equipment. First, some sentential connectives - just the connectives, not their usual (standard or classical) interpretations; they need not (all) even be truth-functional. (On this last point, ask your favorite intuitionist for further exciting details.) A countable number of enumerable sets of variables of different types (individual variables only in the case of first-order quantificational languages; function variables, sentence variables, predicate variables in higher-order languages); one or two variable-binding operators which take formulae into formulae (again just these, not their standard interpretation; on this point, if you didn't believe the intuitionist, ask a substitutionalist); (optionally) infinite stocks of non-logical constants of various sorts; (typically) two delimiters, one for each end; last, but not least, the standard recursive definition of a well-formed formula.

In other words, when you buy a formal language, a language is all you get. Of course it must also be said that there is a standard and classical recipe for concocting a semantics for any language of the standard quantificational variety - the recipe is provided by Tarski in "The Concept of Truth in Formalized Languages" (in [24]). [But as we will note in section IV.C, to say that there is a theory of interpretation for such languages which can be considered standard is not to say that there is only one such scheme available.]

So far all does seem, as promised, peachy. But now an interesting complication rears its head. First, let us assume that much of the work in AI that exploits either logical formalisms or graph-theoretic transcriptions of these is directed toward the representation of the meaning of sentences of a natural language. As it happens, vanishingly little current work in the formal semantics of natural languages is being done within the framework of "standard" quantificational languages. Instead almost all such work goes on within the framework of Church-Henkin type-theoretic (or lambda-categorial) languages [8, 12], usually beefed up with various intensional (including modal) operators.

There are a number of important differences between the Church-Henkin languages and standard quantificational languages; the former allow lambda-abstraction (typically as a device which takes a variable of type β and a wff of type α and forms a wff of type $(\alpha\beta)$, where this is the type of functions from things of type β to things of type α. Church-Henkin languages indeed usually have lambda-abstraction as the sole variable-binding operator - the quantifiers reappear as operations on bound lambda expressions. Such languages are (can be) application-based - the fundamental operation, both syntactically and semantically, being the application of functions to arguments. None of these are characteristics of "standard" quantificational formalisms.

To return to our point about the use of a logical formalism for the encoding of sentences of a natural language: as in philosophy, a significant portion of the most interesting work in AI in this area also adopts, explicitly or implicitly, non-standard formalisms. For example, and explicitly:

> In the following sections I will develop a network representation which permits the use of n-ary predicates..., logical connectives, unrestricted quantification (including quantification over predicates), lambda abstraction, and modal operators such as belief and counterfactual implication...All extensions are analogues of standard notational devices employed in various first-order or higher-order Predicate Calculi. Although no formal semantics are given for the network notation, its correspondence to standard logical notation indicates how such semantics could be formulated. [22, pp. 164-5]

But note that what seems to be taken for granted here is

Assumption 1:

> That there are standard ways of extending first-order languages in the directions alluded to; and

Assumption 2:

> that there are well-developed semantic accounts of these extended formalisms (and/or that there is a clearly "indicated" path on which to proceed in this regard).

Further, "We have developed a network representation for propositional knowledge that we believe to be capable of encoding any proposition expressible in natural language...Its syntax is closely modeled on predicate calculus..." [23, p. 121]. Here, a third assumption appears:

Assumption 3:

> That there is a fairly straightforward encoding of arbitrary sentences of a natural language into such a formalism, which captures the logical form of such sentences.

As regards Assumption 1, we shall here simply note that it is at best misleading to think of the Church-Henkin type-theoretic languages as extensions of standard, even higher-order quantificational formalisms. (In the next section, we shall address ourselves briefly to issues that arise in the context of a fairly minimal "extension" of standard first-order languages to accommodate a particular use of lambda-abstraction.) As regards Assumption 2, we can do no better here than quote John Mylopoulos (we shall, however, return to this issue in section IV.C):

> Another strength of logical schemes is the availability of a clean, well understood and well-accepted formal semantics, at least for "pure" logical schemes that are quite close to First Order Logic. As one moves to representation schemes that try to deal with knowledge acquisition...., beliefs.... and

---

*A minor point of terminology: the use of the phrase "predicate calculus" (or "predicate calculus formalisms") can be a mite confusing. It would be best, we think, to reserve the word "calculus" for inferential relationships defined over a language and not for the language itself.

defaults....the availability of a clean formal semantics becomes more problematic and is an area of active research. [7, p. 6]

While Mylopoulos has somewhat different problems in mind, we certainly share his skepticism about the ready availability of "clean, well understood and well-accepted" theories of interpretation for the kind of formalisms that a number of AI researchers seem to have in mind.* Finally, as to Assumption 3, see section IV.D.

B. On Equivalence of Expressive Power

We want now to take a brief look at an extension of the standard form of first-order languages to allow for the formation, via a form of lambda-abstraction, of complex predicates. Remember that the standard equipment includes no operation for forming complex predicates from atomic ones. To see this, one must bear in mind the distinction between (a) compound open formulae, formed out of atomic open formulae by, say, conjunction (as in "Fx & Gx") and (b) complex predicates, formed by abstraction out of atomic predicates, which can be applied to individual terms (as in "lambda x. [Fx & Gx]"). It is only (a) above that we get for free in a standard first-order language.

So let us imagine that we do have complex-predicate forming operators. What does this extension buy us? Given the standard semantics for first-order languages and the semantics for the new language gotten by minimum mutilation therefrom, one can show that the two languages are equivalent in expressive power - in one sense. But not perhaps in other, at least equally important, senses. The two languages, call them FOL and FOL+, are equivalent in the following way: for every closed sentence of one, there is a sentence of the other (its translation), which has the same truth-value relative to any given model (and assignment function). But do the two languages have the same power with respect to expressing distinct thoughts in distinct ways? Do they have the same power to mark distinctions as to meaning by way of distinguishing among the logical forms of sentences that might be used to express those meanings? It can be argued that only so long as one stays strictly within the confines of standard first-order logic are the two systems equally powerful in this respect. In particular, if one simply adds definite descriptions as complex singular terms to the two languages, FOL+ can be shown to be capable of making finer distinctions as to logical form than FOL. The same situation obtains if one adds modal operators, and in spades, if one adds both. [For a quick peek at the first and simplest case, fix your attention on those old favorites "The king of France is not bald." vs. "It is not the case that the King of France is bald." Remember that in the standard formalism there is no way of distinguishing the negation of an atomic predicational wff from the predication of the negation of an atomic predicate - indeed, there is no such animal as the latter. There is no straightforward, simple way, that is, to capture the distinction between "((lambda x. [Not[Bald(x)]])(ix.King-of-France(x))" and "(Not [Bald(ix.King-of-France(x)])".]

In fact, even without these additions, one can see some possibilities for distinctions within FOL+ that get lost in FOL. For example, consider (1) "Fa & Ga" vs. (2) "(Lambda x.(Fx & Gx))(a)". Both these sentences of FOL+ get translated into the same sentence of FOL - the one that looks just like (1). There are some good reasons, however, for wanting to distinguish between (1) and (2); further on, in our discussion of prototype-nets, we shall indicate at least one such reason. But there is one general point about expressive power that we would like to make here - a point similar to one often made about equivalence results in the theory of computability. The point is that such results don't (can't), by themselves, constitute arguments for choosing any one (provably equivalent) computational system or representational formalism over another. Nor do they by themselves constitute arguments for the overriding importance of that dimension on which the various systems are in fact proved equivalent. When faced with a choice between systems equivalent in a certain respect, one must look to other - perhaps task-specific - factors which distinguish them.

C. What's in a Logic?

There is a form of Assumption 2 above that also needs to be addressed. It is nice to see some recognition of the distinction between a language and a calculus - between a language and a deductive apparatus defined over the language. But it would be just as nice to see some acknowledgement of the difference between a language and a logic; in particular, say, between first-order languages and "first-order logics". The notion of a first-order language (or, if you like of a standard first-order language) is a purely syntactic notion. A language is a standard first-order language just in case it meets certain syntactic specs. The notion of a first-order logic is that of an account of the relationship of entailment (or the property of validity) defined over sets of sentences of a first-order language.

A given language can have many logics defined over it. Often, when folks talk of first-order logic, it can be assumed they have in mind the standard, classical accounts due mainly to Tarski (et al.). Sadly, it can also often be presumed that they are either unaware of other accounts or are insensitive to the distinction between languages and logics. For as noted, there can be, yea verily there are, different brands of first-order logics. Other accounts include the intuitionist account, the so-called truth-value or substitutional account, the probabilistic account, the various and sundry Relevance accounts, Hintikka's game-theoretic semantics, the sundry many-valued accounts, and Zadeh's fuzzy logic account (not to be confused with the probabilistic account). And there are others.

It is, of course, possible to identify a logic for a language with the set of logical truths - according to that logic - of the language. If, moreover, there is a provably complete axiomatization of the valid wffs of the language - for that logic - then one can "represent" the logic with the system of axioms plus derivation rules or rules of inference. Still, distinct things must be distinguished, right?

While we're at it, there is yet another general distinction to be made - or, if you prefer, another nit to be picked: that between a language with a given logic (theory of interpretation) and theories expressed or expressible in that language (with that logic). No one would confuse a first-order formulation of, e.g. classical mechanics, with "first-order logic"; but people are prone to incorporate other, more universal theories, in their various first-order guises with first-order logic itself. It's been done to set-theory; though, in this case at least, a great and longstanding controversy about the relationship between set theory and quantification theory provides a plausible excuse. It may have been done to modal logic, for instance in the following passage from Newell's "Physical Symbol Systems":

---

*Hence our skepticism about claims for extended network notations such as the following in [23]: "The semantics of the formalism, i.e., the intended meaning of the propositional nets, is clear and self-consistent. This is a result of its correspondence to predicate calculus, from which it inherits formal interpretability". We trust that it is clear that the work of Schubert and colleagues is being singled out precisely because it is both tremendously interesting and theoretically self-conscious and explicit.

> Modal notions, such as possibility and necessity, long handled axiomatically in a way that made their relationship to standard logic (hence universal symbol systems) obscure, now appear to have an appropriate formulation within what is called possible world semantics, which again brings them back within standard logic. [18, p. 177]

Here, Newell seems to be confusing (a) the result of extending the classical semantics of Tarski (et al.) a la Kripke (et al.) to handle the new extended languages (e.g., FOL with modal operators treated as logical operators on a par with the quantifiers and connectives) with (b) a first-order theory - in a standard first-order language - of possible worlds and possible individuals (see [14]). And perhaps, but only perhaps, he is assuming the incorporation of theory (b) into first-order logic itself.

D. A Word on Natural Language

Assumption 3 also raises a controversial point - that of the relationship between representational formalisms and natural language. At various points in this paper we may seem to some to assume that the task at hand (for all hands) is that of formally encoding the logical form of sentences of natural language, and then of providing a semantics for such sentences by way of providing a semantics for their formal surrogates. This is a deceptive appearance. We are interested in "non-natural" representational formalisms in their own right. We have no particular applications in mind; hence, no particular natural language applications in mind, either. We do assume here that, to be of significant use, a representational formalism must have the expressive power of a significant fragment of a natural language (any old one will do - say, English). There are lots of interesting thoughts (propositions, if you like) that, unless appearances are wildly deceiving, can be expressed in natural language. They can, one must hope, also be expressed in your favorite representational formalism, even if, compositionally speaking, quite differently. (And one needn't give a damn about those differences.) However we are by no means assuming that the one and only - or even most crucial - use of representation formalisms is to encode the meanings of natural language sentences. It is certainly likely that there are linguistically inexpressible representational structures, e.g., visual and other sensory data.

V. ON PROTO-FORMALISMS

A. Logical Implications of Prototypes

We now turn our attention, as promised, to net work in a less logical vein. In particular, we examine current work most faithful to the psychological origins of network schemes. Arguably the most important legacy handed down from psychologists to network-theorists has been the notion of prototypes and of stereotypical and plausible reasoning (see the work of Rosch and associates [20]). It matters not whether the AI researchers in question knew of the relevant psychological literature directly, merely that their work was, at least in part, informed by a common-sense prototype-type theory. Fahlman puts the relevant point as follows:

> I should also point out that a *TYPE node of the normal sort describes the typical member of a set, but does not define that set. It is not the case that any individual fitting the *TYPE-node's description must be placed in the set. The recognition system is allowed to examine *TYPE-node descriptions and to suggest which type-set a new individual fits into best, but this works by a sort of weighted average over the available features, not by satisfying a formal definition. The *TYPE-node sets are very similar in spirit to the exemplar-based sets that, according to Rosch, dominate much of human recognition and thinking. [9, p. 92]

Now Fahlman is surely not presenting NETL as a mere two-dimensional notational variant of FOL or of any standard quantificational formalism. NETL is a representational scheme in its own right. Even some among the logicians (see [11, 19]) seem to believe that if there is one "representational" feature of most semantic network schemes that is not easily or directly accommodatable within the standard formalisms, it involves prototypes and defaults.**

It is not clear, however, that this supposed difference is one of representation at all (recall the distinctions of section IV). It is at least arguable that, if one wants to realize the "logic of default reasoning", then one need not tamper at all with (the syntax or) the semantics of the representational formalism. Rather, one can tamper a whole lot with the inferential apparatus. (See articles by Reiter and McCarthy in [1]. See also [13].)

On the other hand, one may choose to fool around with the "language" itself, and thence with the logic. In particular one may take as one's base a standard first-order language and add a special intensional (modal) operator to it (see [15] as regards the problem of specifying a semantics for this operator). Indeed, one can even think of Fahlman's *TYPE-nodes as follows: take a Doyle-McDermott-type sentence-forming operator (one applicable to sentences both open and closed) and transform it - voila - into a complex predicate-forming operator, taking predicates into predicates. Given that - in the context of representing the content of sentences of a natural language - the typical operands are, syntactically speaking, common-nouns, this operator acts like a determiner which when combined with a common noun yields a generalized quantifier. In other cases, as when applied to a verb-phrase, it plays the role of an adverb, in particular, an adverb of quantification.

At this point, we must admit to the following qualm: we are at a loss to take sides in a debate as to the adequacy of FOL for prototypes and defaults, since it isn't clear what the sides are. Maybe prototype/default notions are representational ones, and maybe they're not. In particular, to the extent to which theories about psychological processes involved in recognition and subsumption are mixed up with issues of logical form and meaning, it is hard to tell what the intuitive, informal semantics of a formalism really come to. To that extent also is it hard to gauge the applicability of proposed formal-semantical approaches.

---

*Needless to say there are "fragments" of NETL that correspond quite directly to bits and pieces of more standard systems; but even here appearances can be deceiving. For instance, an *EVERY node, it seems, can only govern a compound open sentence. Once one has fixed an (intuitive) interpretation for the primitive *TYPE nodes, one is debarred from linking a single *TYPE-node up to an *EVERY-node. So don't confuse an *EVERY-node with the universal quantifier. (We are not implying that Fahlman himself makes this mis-identification).

**Notice that this feature has nothing to do with the network structure - it has merely to do with the primacy of prototypes.

Regardless of their relation to standard logics, prototype representations are a fact of life in Boat AI representation frameworks. Here we take a quiok look at how prototype representations seem to rule out, a fortiori, the possibility of compound concepts, thereby, and despite illusions to the contrary, limiting themselves to primitive, atomic concepts.•

The principal quantlfioatlonal Import of a prototype is simply this: unless otherwise told, assume that all properties of the prototype hold of eaoh and every "instance" of the prototype. So, if we know CLYDE to be, say. a CAMEL, then we assume that all properties of the TYPICAL-CAMEL hold of him, too. At some point we may learn of some special feature of Clyde that distinguishes him from the prototype (say, for instanoe, that he talks to his^ owner). He simply notate this by "cancelling" the normally inherited property (e.g., that camels don't talk) and substituting the new one. Prototype notations usually carry some explicit cancelling mechanism, which allows them to aooommodate the fact that rarely do real camels match their prototypes exaotly.

Thus, properties of prototypes are always <u>default</u> properties.** Properties are almost universally represented by "slots" of frames or whatever, and the attribution of properties of the prototype to any individual is achieved by "inheritance of properties". In more logical notation, the meaning of a frame representing the concept C, with slot-relationships $R_1, \ldots, R_n$ becomes the following [11]:

$$\forall x\ C(x) \supset R_1(x, f_1(x))$$

$$\wedge\ \forall x\ C(x) \supset R_2(x, f_2(x))$$

ft .

This logical notation expresses the inheritance, but not the default nature of it. The default rules can be expressed as in Relter's "Logic for Default Reasoning" [1], leaving the object language aa is.

Now, what does that leave us with as the import of the frames, units, etc.? First and foremost we have the universal quantification over instances, with embedded existentials for the slots: if Clyde is known to be a CAMEL, then he inherits all of the properties associated with the TYPICAL-CAMEL. Second, we have the fact that since TYPICAL-CAMEL is a prototype, Clyde, as individual camel, has the right to have properties incompatible with those of the typical camel. All this seems quite reasonable from a psychological point of view (again, see [20]).

Given that the properties of the prototype can be violated by instances of it, these properties are clearly non-definitional. This conclusion is reinforced by the "outwsrd" nature of the slots of the frames: if Clyde is (known to be) a camel, then he has typical-oamel-properties; not the other way around (i.e., the connective in the above logical reformulation is the conditional, not the biconditional). Again, this seems well and good, since there are certainly no defining properties for camelhood - the camel is a "natural kind". And, you might add, so are most, if not all of the conoepts that an AI system will have to deal with; leave abstract and defined concepts like RHOMBUS to the mathematicians, and leave the philosophers to argue about whether "bachelor" can be defined.

This intuitively appealing and pervasive line is predicated on an interesting, though unargued and plausibly erroneous, assumption: as the camel goes, so goes everykind else. The unwarranted belief that, with a few technical exceptions, every concept is natural kind-ish has had two significant consequences. The first is an assumption about the shape of the networks that will result when we "represent knowledge". To wit: "An important point about the hierarchies we will want to use is that, while they may be very bushy, they are never very deep" [17, p. 8], and "Knowledge bases consist mostly of short, bushy trees" [9, p. 11]. The feeling seems to be that the structure of the network will be dictated strictly by the relevant natural kind hierarchy (see [17] for further elaboration of this assumption), and the depth of the hierarchy certainly won't exceed that of the "taxonomy of animals". In networks with no composite concepts, this might well be true. However, from a formal <u>representation</u> standpoint - and not a psychological one - there is absolutely nothing to determine the depth of the network, except the grain of descriptions. Conceptual specialization is possible along any dimension of a concept whatsoever, e.g., ELEPHANT-WITH-THREE-LEGS, ELEPHANT-WITH-THREE-LEGS-COLORED-GRAY, ELEPHANT-WITH-THREE-LEGS-COLORED-GREY-LIVING-IN-DALLAS, etc., etc., etc.

The second, and more important, consequence is that there has been no felt need to provide a facility for expressing analytic or definitional connections. This raises no problems with the conceptual counterparts of lexical items like "camel". But just as we can create the English phrase, "camel that talks to the sheik of the burning sand", we should expect to be able to create the node for the composite concept that it expresses. Two things are certain - a camel that talks to its owner can't fail to be a camel, and it can't help but talk to its owner! That is, the composite concept certainly stands in an analytic relationship to its "head" concept, even if that concept is associated with a natural kind.

Since the "Frames" paper, defaults have been almost universally adopted at the expense of definitions or other analytic connections. But this has left network representations in a funny state: inheritance of properties works okay, and one can represent exceptions (three-legged elephants and the like); but one can't represent even the simplest of conceptual composites. And it follows from the lack of composites that every single node (or frame or whatever) in the network is in fact semantically simple - in other words, a primitive (but cf. the Note on "EVERY-nodes, above). An AI system can certainly use such a network as a database repository for such classificatory facts as the user sees fit to tell it (e.g., CLYDE isa TYPICAL-CAMEL), but it cannot draw any such conclusions itself. Without being told <u>explicitly</u>, the system cannot even tell that an elephant with three legs is an elephant!

C. <u>On Simplicity and Definability: Lexical vs. Structural Semantics</u>

With respect to the points made above about lexical items and (non)definability, we have noted a nasty rumor going around to the effect that if one embraces the paradigm of formal (model-theoretic) semantics for natural languages, one thereby, and ineluctably, finds oneself stuck with "analytic definitions" for everything in sight. This rumor is completely unfounded; worse, it is just plain false. Moreover, given what we've said elsewhere in this paper, we feel it incumbent on ourselves to defend the honor, at least in this regard, of formal semantics - even in an orthodox form.

Let's do a worst case analysis - in fact, an almost only case analysis; namely, the application of a rich model-theoretic analysis of higher-order intensional logic to a natural language (English), by a logician who couldn't have cared less about

•This section briefly recapitulates a more detailed discussion that can be found in [5J.

••At least this is the most reasonable interpretation. Sometimes it is totally unclear what Is meant by "prototype".

psychological reality or computational issues, i.e., Montague.* In Montaguesque semantics, each and every lexical item in a natural language is associated with a primitive of (some one of a certain family of) higher-order intensional logics. For such items there is no lexical decomposition; there are no definable lexical items. Of course, syntactically composite items (e.g., "camel that talks to its owner") can have analytic definitions associated with them. But Montague absolutely rejects the line that assigns to lexical items some analysis in terms of "features" or that posits an analysis into some composition of allegedly universal, innate and ultimate conceptual primitives. Mind you, it is easy to take care of those relatively rare lexical items of English (e.g., "rhombus" and "bachelor") which do seem to be analytically definable, by way of what Montague, following Carnap, calls "meaning postulates", which - canonically at least - take the form of universalized conditionals (not biconditionals) with the necessity operator standing guard in front. In sum, there is no necessary connection between formal-semantical analyses of language and the multiplication of (analytic) definitions beyond necessity.

VI. CONCLUDING REMARKS - AN END TO THIS FOL-ISHNESS

TO BE SKEPTICAL OF THE CLAIMS MADE FOR STANDARD FIRST-ORDER LANGUAGES AND THEIR STANDARD LOGICS IS NOT, IPSO FACTO, TO BE ANTI-LOGICAL. Though they may think that with friends like us, who needs enemies?, we in fact intend to be taken as allies of those researchers in AI who have stressed the need for the kind of conceptual clarity that the theoretically self-conscious use of formal languages enforces. But as we shall duly note, we also align ourselves with those philosophers and logicians who look at least a little askance at efforts to force everything and anything into the procrustean bed of first-order languages and their classical logic.

With respect to the above, a little - a very little - intellectual history might be in order. From the very beginnings of AI, there has been a debate about the relevance and usefulness of work in formal, but non-programming, languages done by philosophers and logicians - in particular about the relevance and applicability of work done in and on the fairly standard quantificational formalisms. (We gather this disagreement was in the open at the founding Dartmouth Conference - 1956 - with McCarthy on the side of the "angels"; Minsky and Newell and Simon on the other.) The debate has taken on various forms: procedural vs. declarative encodings of knowledge; general theorem-proving techniques vs. domain-specific heuristics and ad hoc procedures, etc. Now, imagine yourself having fought and having to continue to fight this battle, often against people who seem to be laboring under significant misconceptions as to the nature of formal languages and formal semantics (see [10], which is directed explicitly against some such misunderstandings). To imagine this is to imagine a particular brand of opponent, not one who wishes to counterpose to standard first-order formalisms, e.g., full omega-order type theory with lambda-abstraction, intensional operators, and various other bells and whistles. Rather, the "enemy" doesn't see the point at all of precisely specifying a general formalism for which a well-defined semantics can be given. In this situation, it makes perfect sense to stress the considerable virtues of standard first-order formalisms and their standard Tarski-type semantics, not least among which is that such systems have been studied to a fare-thee-well. We might be accused, then, of systematically misreading and distorting the arguments of the "logicians", distorting them by reading them as if they were directed against a very different type of opponent than that toward which they were in fact addressed. We plead guilty, sort of, but our intentions are honorable. Moreover, it is at least possible that some of the researchers in question have been trapped by the tactics and strategy of their arguments with the anti-logicians into a too ready accommodation to the familiar, the tried and (lest it be forgot) - as far as it goes - the true.

It will doubtless be objected that the present discussion has been merely negative, or at best admonitory, in character. What constructive proposals have we to make? First, we advocate pursuit of alternative styles of formal systems, with careful attention paid to semantics for any non-"classical" constructs. Our feelings on this type of research program are elaborated in [4] and [6].

One not terribly ambitious additional suggestion is that researchers take a look at those approaches to formal semantics that are based on the lambda-calculus-type framework. It has not escaped our attention that (1) the kind of formalisms we here allude to are of proven utility in the semantics of programming languages (vide the work of Scott, et al.); and that (2) they can themselves be thought of as computational formalisms (for which various interpreters can be defined). So, one thing we would like to see is a unification of programming language and representation language research, a unification plausibly less artificial than that proposed by PROLOG proponents, and one in which more of the "give" is from the representational side of the great divide.

That anything would come of such a unification is highly speculative, of course. But now is no time to be a stick-in-the-mud. Witness Jon Barwise:

> One of my own motivations was to use the insights of generalized recursion theory to find a computationally plausible alternative to Montague grammar...[Mine] is a mathematical theory of linguistic meaning, one that replaces the view of the connection between language and the world at the heart of Tarski-style semantics...[It] rejects a level of "logical form" that has anything to do with first-order logic. [2]

Now one of Barwise's complaints against the Montaguesque framework is that it is "computationally intractable". Whatever this complaint might come to, we can't help noting that the recursion-theoretic framework is the natural home for the work on the denotational semantics of programming languages alluded to above.

An even more radical view of standard logic is espoused by the logician, Richard Routley:

*As to its being the only case: Montague-style formal semantics has - so far, and at the very least - the following advantage over all its competitors in the field of well-developed semantic theories for significant fragments of (a) natural language: it exists and they don't. Surely, that is something in its favor. But see [2] and [3].

On the whole there has been far too much effort expended on trying to accommodate philosophical clarifications to going logical systems and straitjackets... - rather than trying to develop logical systems to handle the evident data and to deal with going philosophical problems. Classical logic, although once and briefly an instrument of liberation and clarification in philosophy and mathematics, has, in becoming entrenched, become rigid, resistant to change and highly conservative, and so has become an oppressive and stultifying influence... Classical logic is, as now enforced, a reactionary doctrine. [21]

Finally, and in summary, a word from Mao Ze Dong: "Let at least a few flowers bloom."

REFERENCES

1. Special Issue on Non-Monotonic Logic. Artificial Intelligence 13, 1,2, April 1980.

2. Barwise, J. Some Computational Aspects of Situation Semantics (Abstract). Unpublished manuscript, 1981.

3. Barwise, J., and Perry, J. The Situation Underground. Unpublished manuscript, 1980.

4. Brachman, R. J. On the Epistemological Status of Semantic Networks. In Associative Networks: Representation and Use of Knowledge by Computers, Findler, N. V., (Ed.), New York, 1979, 3-50.

5. Brachman, R. J. "I lied about the trees". Unpublished manuscript, 1980.

6. Brachman, R. J. An Introduction to KL-ONE. In Research in Natural Language Understanding, Annual Report (1 Sept. 78 - 31 Aug. 79), Brachman, R. J., et al., (Eds.), Cambridge, MA, 1980, 13-46.

7. Brodie, M. L., and Zilles, S. N., eds. Proceedings of the Workshop on Data Abstraction, Databases and Conceptual Modelling. SIGART Newsletter 74, January 1981.

8. Church, A. A Formulation of the Simple Theory of Types. J. Symbolic Logic 5, 1940, 56-68.

9. Fahlman, S. E. NETL: A System for Representing and Using Real-World Knowledge. M.I.T. Press, Cambridge, MA, 1979.

10. Hayes, P. J. In Defence of Logic. Proc. IJCAI-77, Cambridge, MA, 1977, 559-565.

11. Hayes, P. J. The Logic of Frames. In Frame Conceptions and Text Understanding, Metzing, D., (Ed.), Berlin, 1979, 46-61.

12. Henkin, L. Completeness in the Theory of Types. Journal of Symbolic Logic 15, 1950, 81-91.

13. Israel, D. J. What's Wrong with Non-Monotonic Logic? Proc. 1st Natl. Conf. on Artificial Intelligence, Stanford, CA, 1980, 99-101.

14. Lewis, D. Counterpart Theory and Quantified Modal Logic. J. Philosophy 65, 1968, 113-126.

15. McDermott, D. Non-Monotonic Logic II. 174, Yale University, February, 1980.

16. Minsky, M. A Framework for Representing Knowledge. In The Psychology of Computer Vision, Winston, P. H., (Ed.), New York, 1975, 211-277.

17. Moore, R. C. A Serial Scheme for the Inheritance of Properties. SIGART Newsletter 53, 1975, 8-9.

18. Newell, A. Physical Symbol Systems. Cognitive Science 4, 2, 1980, 135-183.

19. Reiter, R. On Reasoning by Default. Proc. TINLAP-2, Univ. of Illinois, July, 1978, 210-218.

20. Rosch, E. Cognitive Representations of Semantic Catgeories. J. Experimental Psychology: General 104, 1975, 192-233.

21. Routley, R. Untitled. Unpublished manuscript.

22. Schubert, L. K. Extending the Expressive Power of Semantic Networks. Artificial Intelligence 7, 2, 1976, 163-198.

23. Schubert, L. K., Goebel, R. G., and Cercone, N. J. The Structure and Organization of a Semantic Net for Comprehension and Inference. In Associative Networks: Representation and Use of Knowledge by Computers, Findler, N. V., (Ed.), New York, 1979, 121-175.

24. Tarski, A. Logic, Semantics, Metamathematics. Oxford University Press, Oxford, 1956.

25. Woods, W. A. What's in a Link: Foundations for Semantic Networks. In Representation and Understanding: Studies in Cognitive Science, Bobrow, D. G., and Collins, A., (Eds.), New York, 1975, 35-82.