TAXONOMY+I CLASS

Knowledge Representation and Inference Two

Lecture Eight:

What ISA Is and Isn't: An Analysis of Object Links

and

Lying About Trees: Defaults and Definitions

1 Introductory Comments

These notes are divided into two related parts. The first is about the semantics of the linking relationships used in structured object representation schemes, and the second is about the nature of the objects themselves, and the problems of default representations. Each part is based on a paper by Brachman: part one on *What ISA Is and Isn't: An Analysis of Taxonomic Links in Semantic Networks*, and part two on *"I Lied about the Trees" Or, Defaults and Definitions in Knowledge Representation.*

2 Part 1: What ISA Is and Isn't

2.1 Introduction

Many schemes for representing knowledge are referred to as *structured object* schemes because they feature prominently the notion of an explicit *taxonomic hierarchy*, a tree or lattice-like structure for categorizing classes of objects in the world being represented. The backbone of the hierarchy is provided by some sort of *inheritance* link between the representational objects (*nodes* in some systems, *frames* in others). This link, often called *ISA* (a.k.a. "IS-A", "IS", "SUPERC", "AKO", "SUBSET", etc.), has been perhaps the single most stable element of structured object schemes as they have evolved over the years.

Unfortunately, this stability may be illusory. There are almost as many meanings for the ISA link as there are knowledge representation schemes. Part one of these notes sets out a catalogue of the more common interpretations of ISA, and in so doing, attempts to point out some underlying differences between schemes that, on the surface, appear very similar.

2.2 Some Background

The idea of ISA is quite simple. Early in the history of semantic nets it was observed that much of the representation of the world was concerned with the conceptual relations

expressed in English sentences like "John is a bachelor", and "A dog is a domesticated carnivorous mammal". That is, two of the predominant forms of statements to be handled by knowledge representation schemes were the *predicative* one, expressing that an individual (e.g., John) is of a certain type (e.g. bachelor), and the *universally quantified conditional* one, expressing that one type (e.g., dog) is a subtype of another (e.g., mammal). The easiest way to get such statements into a structured object scheme (or semantic net as it was called in those days) was to have a link that directly represented the "is a" parts of the above sentences, and thus the ISA link was born.

It was quickly noted that the ISA connections formed a hierarchy (or in some cases a lattice) out of the object types being connected — the ISA relation is roughly a partial order. The hierarchical organization made it easy to distribute 'properties' such that shared ones were stored at the place in the hierarchy that covered the maximal subset of objects that shared them. This made the semantic net an efficient storage scheme, since shared properties were not replicated everywhere they were true; they were instead 'inherited' by all objects below the ones where they were stored. This is the notion of *inheritance of properties* that is virtually always mentioned in the same breath as the ISA link.

Once the pattern of a network of ISA links with property inheritance was established, new schemes were developed that used the net as a basis for more elaborate kinds of statements, descriptions, etc. [Bra79] There also arose a debate about whether the network structure was just so much obfuscation of the simple predicative and conditional statements that the ISA links were representing. It seemed that all semantic nets could provide was an indexing facility over formulae — which could just as well (and perhaps better) be expressed in the language of first order predicate logic [Hay77] [Hay79] [Hen79]. The interesting thing about the debate was its 'apples vs. oranges' flavor — each time the logicians tried to pin down the intent of ISA, the net-workers would claim they were missing the point. The same seemed to be true of cross-net comparisons — one scheme was criticized on the basis of what the critic thought the ISA connections should mean, while the scheme was defended on the basis of what the author thought she meant. Not only that, the meaning of the link was often relegated to "what the code does with it" — neither an appropriate notion of semantics, nor one conducive to easily figuring out what the links in fact meant from a knowledge representation point of view.

If nothing else, the various debates over semantic nets have strongly suggested that there is not a single ISA link. Little scientific progress can be made here until we understand at least what the link *could* mean, since a coherent debate on the merits of logic vs. semantic nets cannot be had without the latter making some firm logical claim about the import of ISA. (At the very least, it is hard to imagine trying to justify the advantage of a structured object scheme over logic without formally characterizing the expressive power of the former).

So the question investigated here is: just what is it that ISA links are intended to mean? In the course of this investigation we find some interesting and perhaps surprising things, among them:

1. The more or less standard use of ISA (as an indicator of *default* information) brings with it some potentially serious problems. One cannot use a network based

on it to represent complex concepts, and the notion of 'cancellation' that follows from it can wreak havoc with world knowledge.

2. The tight association of inheritance and ISA serves only to confuse already confused matters further, and only by placing inheritance in perspective (it is an implementation issue and not an expressive power one) can we be clear what the claims about ISA really are.

Along the way, a rational reconstruction of the ISA relation is made, and some constructive suggestions as to how the next generation of knowledge representation languages should be structured presented.

2.3 What ISA Is

First we shall attempt to catalogue the various semantic relations that ISA has been used to represent. This is likely not a complete catalogue, and it may even be unfair to certain knowledge representation scheme designers. But a somewhat careful look at the literature will reveal that this is such a murky area that perhaps these transgressions can be excused.

2.3.1 An Enumeration of ISA-Intents

Before enumerating the ISA's, we must consider the kinds of things that the link has been used to relate. This complicates matters, in that structured objects have been variously thought of as representing sets, concepts, kinds, predicates, propositions, prototypes, descriptions, depictions, general terms, individual terms, and individuals (and probably some more things as well). But it also holds the key to understanding the import of the link. One fundamental split that we can make, despite the variety, is that between *generic* and *individual* interpretations of nodes. Roughly speaking, some objects are thought to be descriptions that can apply to many individuals, and some are thought to represent either descriptions applicable to a single individual, or such individuals themselves.

Generic nodes can be more specific or less specific than other generic nodes – this is what gives structured object schemes their structure – while individual nodes tend to be all at the same level of specificity [1]. Thus, all internal objects in the structure are generic, and the leaves are individual. So we can immediately divide the ISA relation into two major subtypes – one relating two generic nodes, and one relating an individual and a generic. For example, if generic nodes are construed as sets and individual nodes as individuals, then we would expect to find an ISA for the subset relation, and one for the membership relation.

[1] Unfortunately, even this is controversial. Some authors distinguish between two kinds of individuals, roughly corresponding to "John" and "John as a child". The latter is sometimes called a "manifestation".

2.3.2 Generic/Generic Relations

When two generic objects are related by an ISA connection, the intent is usually that one is somehow related to, but less general than, the other. We have at least the following kinds of uses for generic/generic relations:

1. **Subset/superset**: when nodes are taken to represent sets, then the connection between two generic nodes represents the *subset* relation. For example, when nukeSubs and submarines are construed to represent the sets of all nuclear-powered submarines and all submarines, respectively, "a nukeSubs is a submarine" means

$$\forall X((X \in \mathsf{nukeSubs}) \supset (X \in \mathsf{submarines})).$$

2. **Generalisation/specialisation**: generalization — a relation between *predicates* — seems to be expressible as a simple conditional. For instance, if submarine is taken to be a generalization of nukeSub, then an ISA between them means

$$\forall X(\mathsf{nukeSub}(X) \supset \mathsf{submarine}(X)).$$

This is the interpretation of the ISA link offered by Hayes, [Hay79] and is probably the most prevalent semantic net connector. We should point out, however, that more needs to be said about the quantifier on the conditional. While Hayes speculates that a universal quantifier is what is meant, schemes whose objects are interpreted as 'prototypes' (or somehow typical generics) embed their conditionals in more unorthodox *defaults* [Rei80]. This point is discussed further in Section 3.2, as it bears on the expressive capability of the language using it [2].

3. **AKO**: "AKO" means "a kind of", and is intended to stand for the relation between "elephant" and "mammal" in "the elephant is a kind of mammal". This shares much with the generalization relation; however, taken seriously (as opposed to considering AKO synonymous with *generalization*) it implies "kind" status for the objects it connects, whereas generalization relates arbitrary predicates. In other words, it would not do to have an AKO link from an object that does not represent a kind (e.g., "a person who just happens to be walking to school right now" is certainly a person, but seems not to be a *kind* of person). Some representation systems have borrowed from psychological studies of natural taxonomies, and have given different status to different levels of categorization.

4. **Conceptual containment**: in some cases, the intent of an ISA connection is not merely to state a generalization, but to express the fact that one description *includes* another. Instead of reading "a triangle is a polygon" as a simple generalization (such that there are triangles and polygons and this happens to be the relation between them), we want to read it as "to be a triangle is to be a polygon with three sides". This is the ISA of lambda-abstraction, wherein one predicate is used in *defining* another.

[2] Another related issue is the presence or absence of a necessity operator – see Section 2.4.4.

5. **Role value restriction:** another relation between generics is the kind intended in "the trunk of an elephant is a cylinder 1.3 meters long". This use of ISA illustrates that some generic terms in representation languages are intended to refer to *roles* (or 'slots') — not just to types [SB82]. The intent here is to say that the filler of a given role (e.g., trunk) must itself be of a certain type. Note that this version of ISA does not contribute to the taxonomy, since it does not relate the main type being talked about to a super- or sub-type.

6. **Set and its characteristic type:** this is also not a taxonomic relationship; it is the one, for example, between the *set* of all elephants and the *concept* of an elephant. It associates the characteristic function of a set (e.g., a 'prototype' in some systems) with that set.

2.3.3 Generic/Individual

The general intent of an ISA connection from an individual to a generic is to state that an individual is describable by some general description. This relation is often called 'instantiation'.

1. **Set membership:** if the generic is construed as a set, then the relation is membership — "Clyde is a elephant" means "Clyde is a member of (the set of) elephants".

2. **Predication:** this is the use of ISA that simply applies a predicate to an individual. It usually involves a type predicate, like dog, or submarine. If the generic is elephant, and the individual is Clyde, then this ISA expresses the fact that:

   elephant(Clyde)

3. **Conceptual containment:** when the individual object is thought of as a structured description, the relation between it and a generic object could be one of conceptual containment. This is the case between, say, "king" and "the king of France" — the former, generic description is used to construct the latter, individual one.

4. **Abstraction:** there is another relation between an individual and a generic that, in a sense, goes in the opposite direction. This is the relation of *abstraction*, wherein a generic type is abstracted into an individual. This use is evidenced in natural language constructs like "the eagle", in "the eagle is an endangered species". The relation holds between the individual, theEagle and the (generic) predicate or type, eagle.

2.3.4 The General Purpose ISA Link

One approach to the plethora of ISA relations that has surfaced from time to time is the "general purpose inheritance link". Since ISA has so many guises, its advocates argue, they are best served by making a "programmable" connection between objects. The user can turn off attributes that she doesn't want inherited by the more specific objects,

and turn on others that she does. Primitives like "PASS", "ADD", "Exclude", and "SUBSTITUTE" [Fox79] give the user extensive flexibility in making her ISA link do what she wants. The semantics of ISA relations constructed this way, however, cannot in general be predicted.

2.4 An Analysis

Given the above representative sample of ISA-imports, we observe that there seem to be several different dimensions along which ISA links can vary. Each of these are briefly described below:

- First, there is the type of conceptual entity that an object can represent (description, set, predicate, ...) — this has a direct effect on the import of the ISA link;
- Second, there is the basic syntactic function of the link — in particular, we contrast a *sentence-forming* intent with a *description-forming* one;
- Third, for sentence-forming ISA's we have a notion of the "*quantifier*" of the statement (e.g., universal vs. default);
- For these types of relations, we need also consider *modality* (necessity vs. contingency);
- Finally, we must consider whether or not the link, by its very presence in a network, makes an *assertion*.

2.4.1 Effects of Object-Meaning

The first major influence on the meaning of the ISA relation is the types of the objects that ISA is relating. For example, if ISA is a relation between two sets, then it is usually about their *membership* (or possibly cardinality) — typically *subset*. The generic/individual version is usually the set membership relation (although there are variations [Hen79]).

In contrast, when the items to be related are predicates, then ISA typically has something to say about predications that *follow* from other predications, using the material conditional (e.g., "a whale is a mammal" corresponds to $\forall X(\text{whale}(X) \in \text{mammal}(X))$). The hierarchy derived from this style of ISA has an "if ...*then*..." flavor — if something is a whale, *then* it is a mammal, etc. [3]. When the ISA-related objects are intended to be descriptions, or 'concepts', then the relation between them is either about the *structure of the descriptions* themselves or about the classes of objects satisfying the descriptions. In the former case, an ISA like "a triangle is a polygon" says that part of the description of any triangle is that it is a polygon. The same definition-inclusion relation holds if one of the descriptions is an individual description.

It should be pointed out that this last type of ISA relation — carrying structure between structured descriptions — is one of the points of most radical departure from

[3] Note that this fails to say what it is that makes something a whale in the first place (see Section 3.2 for more on this.)

standard predicate logic-based representation schemes. Almost all of the other ISA sub-factors that we are pointing out are easily expressed in standard quantificational languages.

2.4.2 Sentence-Forming vs. Concept-Forming

ISA links in most structured object schemes are used to make statements about the world. For example, imagine that we had an object that we wanted to stand for the class of Indian elephants, call it "indianElephant". If we wanted to state that these elephants were (typically) brown, we would simply assert "indianElephant ISA brownThing".

This kind of statement-making about independently motivated classes is standard fare for structured objects (semantic nets in particular). However, cleverly named nodes make it easy to gloss over something we cannot do with the above kind of ISA, namely, make indianElephant have indian and elephant as parts of its meaning. There must be another kind of ISA to express this other kind of relation (just having the right lexical sequences as parts of a long hyphenated name buys us nothing; indianElephant could just as well have been called "G0047"). Thus, a basic split between statement-forming and concept-forming ISA's should be made [4].

2.4.3 A Weak Sense of 'Every'

Just about everyone uses the ISA relation to make a statement about a pair of classes or a class and an individual. While the obvious quantifier to assume holding over such statements is the universal one [Hay79], this appears not to be in every structured object designer's mind. In particular, many ISA's are thought to be 'cancellable'. For example, "a bird is a flying-thing" is taken to apply for any bird, as long as it is not explicitly known that that bird cannot fly — "flying" is a *default* property for birds [Rei80]. Sometimes we will want to make this kind of statement, which can have exceptions, and sometimes not (e.g., "a person is a mammal"). So, for any sentence-forming ISA link, we need to know if it is a true universal, or merely a default.

2.4.4 Is This Necessary?

While rarely used in structured object schemes, there is another dimension along which ISA's can vary. Some truths could have been otherwise — this particular circle could have had a radius other than the one that it has. However, the fact that its area is 2π times its radius (no matter what that radius is) could not be otherwise. The latter is a *necessary* truth, the former a *contingent* one. Not only are some ISA's exceptionless they can never be false.

2.4.5 To Assert Or Not To Assert

In many systems, the ISA link asserts a truth by its mere presence. Having the statement "Clyde ISA elephant" in your structured object representation means your

[4] The latter kind of ISA relation is what we have been calling "conceptual containment" above. For more on this dichotomy see the Israel and Brachman reference [IB81].

system 'believes' that fact about Clyde. If this is the only form of ISA relation, we are not free to contemplate a proposition without asserting it. Thus the distinction between asserted ISA's and simply structural ISA's adds another dimension to the ISA connection.

2.4.6 Summary

We might summarize the many meanings of the ISA link in a matrix, as depicted in Table 1. Each row in the table is labelled with the kinds of generics and individuals being related by ISA, and shows correspondingly the meaning of ISA between (1) two generics and (2) a generic and an individual. For example, the second row indicates that when generics are construed as predicates and individuals as constants, then "⟨generic⟩ ISA ⟨generic⟩" is a universally quantified material conditional statement, and "⟨individual⟩ ISA ⟨generic⟩" is a predication (the third to last row is the exception here, since the abstraction ISA goes from a generic to an individual). The last two rows illustrate ISA's between generics of different types (g1 being the subordinate object and g2 the superordinate).

Types of Objects Related by ISA		Type of ISA Link	
generics	individuals	generic/generic	generic/individual
set	individual	SUBSET	MEMBER
predicate	constant	UNIV MATERIAL CONDITIONAL	PREDICATION
'kind'	individual	AKO (SUBKIND)	MEMBERSHIP IN KIND
structured description	individual concept	CONCEPT CONTAINMENT	CONCEPTUAL CONTAINMENT
structured description	individual	—	DESCRIPTION ('FALLING UNDER')
'prototype'	individual	SHARING OF 'TYPICAL' PROPERTIES	SIMILARITY TO PROTOTYPE
'role'	individual	—	SPECIFICATION OF ROLE FILLER
predicate or structured description	'kind'	—	ABSTRACTION
g1: role g2: structured description or predicate or 'prototype'	—	CONSTRAINT ON ROLE FILLER('VALUE RESTRICTION')	—
g1: set g2: 'prototype' or predicate or structured description	—	SET-CHARACTERISTIC FUNCTION	—

Table 1 — Summary of the ISA link.

2.5 A Logical Question

The above analysis has left us with the following picture: there is a fundamental split between the kinds of things we can say with ISA into those that take one concept and form another out of it, and those that make some sort of statement about the relation between two sets or the arguments to two predicates.

The ones that are used to make statements have four sub-components:

1. The *assertional force* of the statement — whether or not the statement represented by the ISA is taken as a statement of fact.
2. The *modality* of the statement — whether the truth represented by ISA is necessary, or is just contingently true (and could thus be contemplated to be otherwise).
3. The *quantifier* of the statement — whether the matrix (see below) is to be considered universally true, or just "true unless explicitly cancelled".
4. The *matrix* — the content of the statement. As illustrated above, this generally has the structure of a set inclusion (or membership, in the case of a generic/generic ISA) or a material conditional (or predication, in the case of a generic/individual ISA) statement.

These four factors used to form the ISA relation look suspiciously like the pieces that make up more or less special cases of more or less standard logical statements (in, say, prenex normal form). Is the ISA link, then, accounted for completely by standard, off-the-shelf logical machinery?

Well, much of it can be — but all of the factors combine to force us out of the realm of the standard, well-understood logics. This point is covered in more depth in [IB81], but the modalities and defaults are enough to put us on tricky logical ground. When lambda abstraction, or something like it (the concept-forming kind of ISA), is added, then a semantically well-specified structured object account might be as reasonable a logic as any other. In addition, there are other factors that make the concept-forming style of ISA and the resultant network-style languages look like real alternatives to standard predicate logic accounts. For one, having structured terms that are interrelated provides a basis for a formal account of the terminology used to describe a domain, whereas standard logics do not support defined, non-atomic predicates [5].

Perhaps most importantly, structured object-style representation emphasizes certain compelling patterns in knowledge representation that do not emerge from predicate-logic based ones. For example, at least one interpretation of the concept/role paradigm can be expressed easily in a standard logical language [Hay79], but that pattern is just one among infinitely many. Structured object schemes have elevated the pattern to the level of a built-in form because of its widespread utility in representing knowledge. Another compelling pattern is the very distinction of "ISA" from "is" — structured object schemes have acknowledged the prevalence of reasoning based on types from very early on, and have made a prominent distinction between the sense of "is" in "John is a man" and all other senses of "is" (e.g., "John is running scared", "John is extremely tall").

[5] While this is not an issue of more expressive power (one can perhaps say what needs to be said in a primitive-predicate-based system), it is a matter of perspicuity, and perhaps even computational tractability. These factors are discussed in more depth in two other papers [BFL83] [BL82].

2.6 What ISA Isn't

One important observation to be made about our analysis of the semantics of the ISA link is that 'inheritance of properties' has played no part in our understanding. This is not without good reason — even though much has been made in the past of the significance of inheritance in structured object schemes, no-one has been able to show that it makes any difference in the expressive power of the system that advertises it. At best, any argument that inheritance is useful is made on pragmatic grounds: it saves storage space in an implementation (or 'localizes' information to be changed).

Without denying the importance of implementation concerns, the extent to which inheritance is a useful property, is strictly an implementational one and bears no weight in any discussion of the expressive or communicative superiority of structured object representation schemes. For one thing, any expression of properties at "the most general place" in a network-style system can be duplicated easily in a logical one. We simply associate the property axioms with the most general predicate, and the standard conditionals do the rest. Further, inheritance is only one cut at the time/space tradeoff for storing properties in a structured object scheme; it may be tremendously easier in some cases to store all properties explicitly where they apply to cut down search time. Thus, although the ISA relation can be factored into sub-components, the useful ones for semantic purposes are assertional force, modality, etc., and not "pass this property" and "block this one". While these latter may be very useful tools for implementing a particular ISA methodology, they should not encroach on discussions of the adequacy of structured object schemes for representing knowledge.

2.7 What ISA Ought To Be

What might be a viable prescription for future ISA-schemes? It is reasonable that the obvious one should be explored first. This section briefly reiterates a strategy for designing representation systems that is presented in depth elsewhere [BL82].

First, we should carefully distinguish between description- or term-forming operators and sentence-forming ones. There is a useful place for each, and a marriage of the traditional logical approach and the more recent "object-centered" terminological approach along these lines should be explored. Structured predicates (or concepts) play an important role in expressing knowledge, and technical vocabulary should be preserved in a representation, despite the fact that all statements using defined predicates might be reducible to a set of statements using only primitive ones. One way to do this would be to have a network-style representation scheme where the principal relation is the ISA of conceptual containment completely distinct from a network (or set of axioms) expressing the facts of the world. The latter set of statements (in the "assertional component") would use terminology from the former (the "terminological component").

The assertional component is where statements about the world are made. Thus, it needs to have the expressive and inferential power of at least standard predicate logic. This could be accomplished by using a standard quantificational language, or we could use a more network-like language. In the latter case, the backbone of the network would be the sentence-forming style of ISA.

This ISA could be broken down componentially into a "prefix" and a "matrix". The prefix would have three parts: the assertional force of the statement (whether or not the statement is taken to be true), the modality (necessary, etc.), and the quantifier (universal, existential, "typical", etc.). The matrix itself would include conditionals like those discussed above. What needs to be done is to specify what is implied for the assertional component by each kind of structure in the terminological component.

2.8 Some Comments

Structured object schemes have prospered as an approach for knowledge representation, but all the while, their keystone construct — the ISA link — has wavered considerably in its interpretation. ISA has been used principally to form sentences that could be asserted — in particular, sentences with a default import. However, there are many other things that ISA has been used to mean, and comparison between different structured object schemes and between structured object schemes and logic has been rendered all but impossible. The analysis presented above indicates that things might be a lot clearer if ISA were broken down into its semantic sub-components, and those used as the primitives of a representation system.

3 Part Two: Lying About Trees

In this part of the notes some problems concerning the semantic interpretation of structured object systems which embody notions of *typical*, *prototype*, *defaults*, and *property cancellation* are outlined.

3.1 Clyde's Revenge

Clyde the elephant has been the subject of much definition and dissection by the knowledge representation community over the years, as is evidenced by the literature. But, it seems that the elephants may have the last laugh. Clyde's revenge comes in the form of a well known elephant joke, that goes something like this:

Q: What's big and grey, has a trunk, and lives in the trees?

A: An elephant — I lied about the trees.

Suprisingly, "lying" about properties, such as where elephants live, seems to be basic feature of many structured object knowledge representation schemes, particularly those described as *frame-based* systems, which employ *inheritance of properties* and *cancellation* of inherited properties. Though such notions enable *default* or *prototype* descriptions to be represented, which can be suitably modified to fit particular instances, they also limit the interpretation of such objects to only defining strictly default conditions. They therefore cannot represent definitional conditions, or even contingent universal ones. Without some definitional capability, such representation schemes cannot express even

simple *composite descriptions*, like "elephant whose colour is grey", or "polygon with four sides".

3.2 The Default Interpretation of Objects and Links

As mentioned earlier, Hayes [Hay79] suggests that the material conditional (if...then...) is the connective represented by ISA between objects. However, the most prevalent use of ISA seems in fact to be as a default — "bird ISA flyingThing" is usually taken to be a truth about birds until it is explicitly retracted ('cancelled'). The motivation for this approach is simple — without the ability to cancel object properties in general, exceptions cannot be represented, and the world is such that exceptions are an important aspect of knowledge representation. Unfortunately, while the default view of objects and links allows structured object schemes to deal with an exception-full world, there are some important things that it keeps us from being able to express.

In particular, one consequence of the default view is that the objects — intuitive hyphenated names notwithstanding — cannot be thought of as representing the concepts that their names suggest. Instead, they must be considered simply as holding points for bundles of default properties. This is because of the strictly one-way nature and cancellability of the object properties expressed by the default ISA links: *if* Clyde is an elephant, *then* he has the properties typical of elephants. This rule does not work the other way around — if Clyde has typical elephant properties, we can *not* conclude that he is an elephant, since he could be just about anything, say a giraffe, with none of the typical giraffe properties (all cancelled) and exactly those properties typical of elephants. Thus an object like elephant doesn't really stand for the concept of an elephant (none of its properties embody the definition of an elephant), but is rather a somewhat strange thing — a collection of *typical-elephant-properties*, or *elephantness*.

The fact that defaults have been almost universally adopted at the expense of definitional object links is perhaps well-intentioned: there are arguably no defining properties for elephanthood (the elephant is a 'natural kind'). And, one might add, so are most, if not all of the concepts that a knowledge representation 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). However, with only defaults we cannot represent even the simplest of conceptual composites, like "elephant with blue eyes" (the best we could do would be to assert that an elephantWithBlueEyes *typically* has blue eyes). A knowledge representation system can use a strictly default-based structure as a database repository for such classificatory facts as the user sees fit to tell it (e.g., Clyde ISA elephant), but it cannot draw any such conclusions itself. Without being told *explicitly*, the system would not even know for sure that an elephant with blue eyes was an elephant.

Incidentally, notice that the implied sense of *typically* is usually taken to mean *in the absence of any information to the contrary, assume ...*, and has nothing to do with frequency of occurrence. In some dialects, *typically* is more closely synonymous with *usually*. The sense in which it is used in structured object schemes means, however, that a *typical* property could be violated in every instance.

3.3 The Myth of Cancellation

The preponderance of default-style objects in structured object schemes has admitted *cancellation of properties* into the realm of representation. With it has unfortunately come a set of technical problems [FTvR81] — but even worse, the semantic consequences of cancellation have hardly begun to be thought through. The intuitive feeling is that cancellation can be constrained to handle just the meaningful cases of exceptions; the truth is that cancellation admits bizarre, unintuitive structures quite easily.

For example, we could easily imagine the node 3leggedElephant appearing below elephant, with the cardinality of the leg-slot changed (assuming that elephants typically have four legs). But we could later decide to cancel that property, and have 4leggedElephant below 3leggedElephant. Since the names are so suggestive, this looks reasonable; however, we should remember that 4leggedElephant is really just a set of properties (not the concept or definition of a four-legged elephant) — in fact, it is the same set as that represented by elephant! And we could go on cancelling and re-cancelling *ad infinitum*. Or, we could easily form structures that say that (1) "an Indian-elephant is a elephant", (2) "Clyde is a Indian-elephant", and (3) "Clyde is *not* an elephant". Arbitrary cancellation is not very constructive.

Another important consequence of this default interpretation of objects is that the slot notation of properties cannot be used, without some kind of modification, to make unequivocal universal statements. Assuming from the elephant frame-object which has a slot *colour* set to grey, does not mean that there could not be an exception to this property made immediately below it or anywhere below it in the representational structure. But we certainly want to be able to make exceptionless universal statements, since there are many domains and situations where properties do hold for all instances, and *must* hold by definition, or as a necessary condition — which it is clear is *not* what standard frame-object slots represent!

3.4 Kinds of Cancellation

When considering cancellation of property values (slot values), we need to ask the question: what is being cancelled, the *value* — having a cylindrical trunk, or the *property* or *attribute* — having a trunk at all? For example, we can say that Clyde has all the properties of the typical elephant declared in the object elephant, except that his trunk is hexagonal. Here we are cancelling just the value of the default trunk slot and replacing it with a new one, hexagonal. But we might also want to say that Clyde has no trunk, in which case we would want to cancel the *property* "has trunk".

This leads on to further kinds of cancellation. It is one thing for Clyde to not have a trunk, but it is quite another for him to not have shape. Since he is a physical object (unless our elephant object is supposed to be describing fictional or 'Cheshire' elephants — like a certain cat) he may be able to lose parts, but he cannot lose properties essential to being a physical object. So there is a distinction to be made between properties which can be cancelled and ones which cannot be, in other words some properties seem to need to be *sacred*.

There are other strange possibilities. A giraffe might be represented as an elephant whose trunk is its neck, and thus inherits the value cylinder for the shape of its neck.

Here cancellation is being used as just a syntactic device. It thus does not seem to be possible to rule out the ability to cancel the attribute name while leaving the value. Indeed it seems to be possible to cancel *every last attribute*, thus leaving us with possibilities like "a rock is an elephant, except that it has no trunk, it isn't large, it has no legs, ...".

If this is beginning to give you the idea that our object elephant does not really represent the concept of an elephant, then you would be right. In any case, if we are going to admit the use of cancellation at all, then we are going to have to sort out ways of dealing with the problems outlined above first.

3.5 Sufficiency and Compositionality

Given the default interpretation of structured object schemes which adopt the standard *frame* notions, and the realisation that ISA links do not represent even simple contingent universals that they might appear to, leads us to suspect that even stronger statements like definitional descriptions are out of the question. This might at first not seem to be a problem, but imagine trying to define two potentially similar frame-objects quadrilateral and elephant. Where something is a quadrilateral if and only if it is a polygon and has four sides, which seems unproblematic. And where elephants are mammals and they have four legs, which is not so straight forward — even it it were necessarily the case, we could not conclude that any four-legged mammal was necessarily an elephant. That is, regardless of whether or not the complex property, *four-legged mammal*, is necessary, it is certainly not *sufficient* for being an elephant. In fact, it is strongly believed that *no* combination of properties is sufficient to capture what it means to be an elephant — in other words, *natural kind* concepts cannot be defined. In contrast, there is nothing more to the story of quadrilaterals than *four-sideness* together with *polygonicity*.

Thus many structured object representation schemes strongly favour the non-mathematical domains, perhaps with good reason: why worry about definition if, at best, only quadrilaterals and the like can be defined? Well, consider this: once we have the concept of an elephant — the natural kind, primitive, or whatever — from it we can construct an indefinite number of composite concepts (objects), each of which is in a relation to elephant that is surely definitional. For example, the concept of an elephant with three legs — called our frame-object for it "elephantWithThreeLegs" — is a simple composition of two attributes, each of which is necessary, and the pair of which is sufficient. That is, it is impossible to have an elephant with three legs that is not an elephant, and it should be impossible for an object that both is an elephant and which has (exactly) three legs to fail to be counted as an elephantWithThreeLegs, and it is undeniable that frames like elephantWithThreeLegs will be useful.

The point that comes out of this analysis is that in order for a knowledge representation system to be able to handle any reasonable range of descriptions — *even the simplest constructed from natural kind-like concepts* — some type of definitional (i.e., compositional — not "typical" kind) structuring capability is necessary. To form descriptions of a very common sort, necessity and sufficiency are required. Put another way, any knowledge representation inference mechanism worth having must be able to deduce — without fail — that a three-legged elephant has exactly three legs. The internal structure of a non-atomic concept like elephantWithThreeLegs must be transparent to the system's inference mechanism, or else it can't tell if the knowledge-base builder is lying about the object's apparent composition.

Unfortunately, most frame-based and semantic net systems are forced to miss such immediate inferences because of the way they are constructed. A default interpretation of properties is almost always adopted at the expense of any serious definition mechanism. Slots, since they can be overridden, can be taken to represent only properties that *typically* follow from being an instance of an object. So we can't use the properties in both directions in the cases we should be able to — in all cases something has to be explicitly asserted to be under a frame before it can be determined what properties follow from that attribution. And even then, we cannot categorically draw the inference — we can't confidently conclude that Clyde is a mammal even it we have stated that he is an elephant, because such properties are cancellable, and we would be kidding ourselves if we were to assume that the ISA link means truly all elephants are mammals, as we have seen in part one of the notes.

3.6 The Myth of Structure

As a result, our representation system cannot use the structure of the objects to determine whether one is more general than another, even when it should be able to. That is, the typical frame-based system cannot tell if one frame-object is a specialisation of another even if that fact should be transparent from its content. What this all goes to say is, that *every description in such structured object schemes is primitive.* While the so-called "frame-objects" look complex, they act more like atomic primitives, so doomed by the failure to express real universals and sufficient conditions. This leaves the knowledge-base builder having to tell the system *every* immediate superdescription of a newly added description — even a composite one that, by all rights, ought to wear its meaning on its sleeve. In other words, standard frame-objects which make use of default interpretations and cancellation of inherited properties are not structured objects at all; their wholes would seem to be *less* than the suns of their parts.

In a system without true compositional structuring, there is no notion of *representation by structured correspondence.* Only if we are not allowed to lie about properties can the system know automatically. Every composite description should really have a proper place in the network structure based only upon its internal structure. One very important job of representation systems, then, should be to keep the representational object in their places. This gives rise to the notion of a *classifier* — a fundamental part of a knowledge representation system that places objects where they belong with respect to all other previously defined objects. Classification shifts an important burden from the user to the system where it belongs, and it is also very useful in knowledge acquisition and maintenance, as well as enabling the system to make inferences that it really should be able to make. In any case , no matter how hard we want to believe the descriptions in our representations have intrinsic compositional meaning — and no matter how complex we make their names — unless the system can distinguish between defaults and definitions, *they all look the same* to it.

4 Some Concluding Comments

The occasional riddle like the one at the start of section three is enjoyable. But imagine how unfunny it would be if the joke went something like this:

Q: What's big and grey, has a trunk, and lives in the trees?

A: A giraffe — I lied about the colour, the trunk, and the trees.

Or, worse still:

Q: What's big and grey, has a trunk, and lives in the trees?

A: An idea — I lied about the colour, the trunk, the trees, and about the "lives".

If you think this is boring, pity the common frame-based structured object knowledge representation system, which has to live under these conditions. It seems that general arguments about the need to handle exceptions, and about the obvious way to represent them, have led to a breed of knowledge representation schemes that do not allow the interpreter to predict when all bets will be off. While today's AI antilogicians might protest that definitions are impossible or irrelevant — after all, they say, "rules are made to be broken" — the pendulum seems to have swung too far in the direction of the exception-handling. When *all* rules are made to be broken, then they are no longer rules at all. Another way of putting all this is to say that frame-based knowledge representation systems throw out the compositional baby with the definitional bath water.

There are thus four simple ideas to take away from this lecture:

1. The semantics of the ISA link is many and varied, so be careful.
2. Frame-objects are typically not very *frame* like.
3. Definitions are more important than you might think.
4. Cancellation is worse than it looks.

If the analysis presented here makes you think twice before buying next time someone tries to sell you a knowledge representation system that allows cancellation, then you will have appreciated one of the most important points of this course.

5 In Defense of Minsky

What these notes have attempted to set out is some of the problems associated with the use of default objects in structured object representation schemes. It is tempting to lay the blame for these problems at the feet of Marvin Minsky who, in his "Frame" paper is often attributed with the original idea of default objects and cancellation in representation schemes. The attribution of blame in this way would be unfair for two reasons. First Minsky was not the only person to have been talking about the need to represent and reasoning about defaults in order to represent and reason about the real world. And secondly, and perhaps more importantly, Minsky, in his "Frame" paper was more concerned with issues of a more 'Psychological' and 'Philosophy-of-Mind' nature, not so much with presenting a knowledge representation scheme to be used in knowledge-based systems. This can be seen quite clearly from the discussion presented in the later sections of the paper. The problem has been more to do with an over interpretation of the ideas presented by Minsky by those looking for a more convenient and humanly tractable way of representing knowledge in a computer-based system.

For a better idea of what Minsky was talking about and why see Marvin Minsky's book *The Society of Mind*, in which he presents his theory of mind in a novel and effective way.

6 Required Reading

The **Required Reading** to go with this week's notes is Chapter 17, page 309, of the Big Red KR Book, *On Inheritance Hierarchies With Exceptions*, by Etherington and Reiter. This paper serves as a good illustration that solving some of the problems discussed in the notes is not as easy as it might at first seem!

References

[BFL83] R. J. Brachman, R. E. Fikes, and H. J. Levesque. *KRYPTON: A Functional Approach to Knowledge Representation.* FLAIR Technical Report No. 16, Fairchild Laboratory for Artificial Intelligence Research, Palo Alto, CA 94304, May 1983. Also appears in *IEEE Computer, Special Issue on Knowledge Representation*, September 1983.

[BL82] R. J. Brachman and H. J. Levesque. Competence in knowledge representation. In *Proceedings of AAAI-82*, pages 189–192, Pittsburgh, PA, 1982.

[Bra79] R. J. Brachman. On the epistemological status of semantic networks. In N. V. Findler, editor, *Associative Networks: Representation and Use of Knowledge by Computers*, pages 3–50, Academic Press, New York, 1979.

[Bra83] Ronald J. Brachman. *What ISA Is and Isn't.* FLAIR Technical Report No. 15, Fairchild Laboratory for Artificial Intelligence Research, Palo Alto, CA 94304, May 1983. Also appears in *IEEE Computer, Special Issue on Knowledge Representation*, September 1983.

[Bra85] Ronald J. Brachman. "I Lied about the Trees" or, defaults and definitions in knowledge representation. *The AI Magazine*, 80–93, Fall 1985.

[ER85] D. W. Etherington and R. Reiter. On inheritance hierarchies with exceptions. In R. J. Brachman and H. J. Levesque, editors, *Readings in Knowledge Representation*, chapter 17, pages 329–334, Morgan Kaufmann, 1985. Also appears in *Cognitive Science*, Vol. 1, 1977.

[Fox79] Mark S. Fox. On inheritance in knowledge representation. In *Proceedings of IJCAI-79*, pages 282–284, Tokyo, Japan, 1979.

[FTvR81] Scott E. Fahlman, D. S. Touretzky, and W. van Roggen. Cancellation in a parallel semantic network. In *Proceedings of IJCAI-81*, pages 257–263, Vancouver, B.C., 1981.

[Hay77] Patrick J. Hayes. In defence of logic. In *Proceedings of IJCAI-77*, pages 559–565, Cambridge, MA, 1977.

[Hay79] P. J. Hayes. The logic of frames. In D. Metzing, editor, *Frame Conceptions and Text Understanding*, pages 46–61, Walter de Gruyter and Co., Berlin, 1979.

[Hen79] G. G. Hendrix. Encoding knowledge in partitioned networks. In N. V. Findler, editor, *Associative Networks: Representation and Use of Knowledge by Computers*, pages 51–92, Academic Press, New York, 1979.

[IB81] D. J. Israel and R. J. Brachman. Distinctions and confusions: a catalogue raisonne. In *Proceeding of IJCAI-81*, pages 452–459, Vancouver, B.C., 1981.

[Min85a] Marvin Minsky. A framework for representing knowledge. In R. J. Brachman and H. J. Levesque, editors, *Readings in Knowledge Representation*, chapter 12, page 245, Morgan Kaufmann, 1985. Also appears in *Mind Design*, J. Haugeland, editor, MIT Press, 1981.

[Min85b] Marvin Minsky. *The Society of Mind.* William Heinemann Ltd., London, 1985.

[Rei80] Raymond Reiter. A logic for default reasoning. *Artificial Intelligence*, 13(1,2):81–132, 1980.

[SB82] J. G. Schmolze and R. J. Brachman. *Summary of the KL-ONE Language.* FLAIR Technical Report No. 4, Fairchild Laboratory for Artificial Intelligence Research, Palo Alto, CA 94304, May 1982. Also appears in *Proceedings of the 1981 KL-ONE Workshop*, pages

Tim Smithers
March 1988