Concepts, Definitions, and Inheritance

Interpreting the atoms of lexical decomposition

by

DAVID ZORNEK

Inheritance Networks and the GDIT

It should be clear at this point that there is a fairly high degree of analogy between the inheritance relation of decompositional theories and conceptual hierarchy, since both the set-inclusion relation and the inheritance relation are transitive and reflexive, but not symmetric. This is not enough, however, to establish more than an analogy between semantic content and conceptual content, since although reflexivity and transitivity are properties of both inheritance and set-inclusion, set-inclusion is restricted by additional axioms, i.e. the axioms of ZFC, that do not generally apply to type-inheritance. In particular, by the axioms of ZFC, no set can be a member of itself, despite the fact that all sets are included in themselves. However, types don't have members, but *tokens*. The difference is not purely terminological. The closest analog of set-membership in the domain of types is the type-token relation,[1], but all types are tokens of themselves, which implies that the type-token relation *cannot* be the same as set-membership. There are domains over which we can define inheritance and set-inclusion relations that do not map onto one another, and computer scientists have distinguished between inheritance and set-inclusion as distinct relations [?] [?] [?]. It is, therefore, a non-trivial matter to show that inheritance is a model of set-inclusion relations such as conceptual hierarchy and semantic content.

Conceptual hierarchy has already been explained in some level of detail, but semantic contentent remains far too vague a notion to be much use here. First, I will clarify exactly what I take semantic content to be. More precisely: Semantic content simpliciter is probably too broad, general, and complex a notion to straightforwardly identify a formal model of all of its aspects, so I will identify some specific empirical observations about semantic content that are modeled by my framework. I will then define the notion of an *inheritance network*, a partial order over two relations that seem to capture these observations. Once the framework has been set up, I will state the Genus-differentia Inheritance Theorem, which establishes an

[1] The type-token relation is understood here in a mathematical sense. If $\alpha \sqsubseteq \beta$, i.e. if α inherits from β, then we say α is a token of β

isomorphism between inheritance networks, (aspects of) semantic content, and conceptual hierarchy. Finally, I will identify some philosophical implications of the Theorem, as well as some additional ideas that will be useful in carrying out the case study.

It is a well-known feature of word meaning that lexemes can be related to one another by a *lexical entailment* relation, in which the semantic content of one word is regarded by native speakers as being entails in the semantic content of some other word. D. A. Cruse has offered a number of diagnostic tests for lexical entailment. Certain sentence forms will provoke an intuition of "oddness" when lexical entailment relations are violated [**?**]. Two such sentence forms are illustrated in teh follow four sentences:

(1) ? It's a dog, therefore it must be a cat.

(2) It's a dog, therefore it's an animal.

(3) ? It's a dog, but it can bark.

(4) It's a dog, but it can't bark.

(1) and (3) are odd, while (2) and (4) are not. In (3), it seems odd to conjoin "It's a dog" with "It can bark" using "but," because one semantically entails the other. This entailment can be modelled by the inheritance relation. If we set $\mathbf{dog} \sqsubseteq \mathbf{barks}$, then $\alpha : \mathbf{dog} \vdash \alpha : \mathbf{barks}$, which is a straightforward formalization of the entailment exhibited by the oddness of (3). (4) lacks oddness because the meaning of "but" conforms to the fact that non-barking is unexpected behavior for dogs.

Another important kind of semantic entailment is exhibited in (1) and (2). (1) seems odd because "It's a dog" entails "It's not a cat;" we regard "dog" and "cat" to be semantically disjoint. (2) however, is a perfectly non-odd sentence, because all dogs are animals (i.e. $\mathbf{dog} \sqsubseteq$, i.e. $\alpha : \mathbf{dog} \vdash \alpha : \mathbf{animal}$), which means that "dog" and "animal" cannot be semantically disjoint. No relation has been explicitly introduced thus far which can be used to model semantic disjointness, although the dotted lines in Figure 1, Chapter 2, do represent such a relation, which will be defined explicitly below.

Not all *but*- and *therefore*-sentences tell us about semantic content—most of the ones people actually use don't. In fact, (2) and (4) do not. Nor do such sentences as "He's a Repbulican, but he's a pretty nice guy," or "It's a Chevy, but it gets pretty good mileage." Only sentences like (1) and (3), which do provoke native speakers to have intuitions of oddness are taken by Cruse as diagnostics for semantically important relations between words.

There are other semantic relations we might want to model; if so, we will want to enrich the framework presented below by adding additional relations to our partial order. Adding more relations will add more complications, however, and for the time being I simplify matters by including only those relations that are necessary to establish the isomorphism to concepts.

1 Inheritance networks and genus-differentia definition

I will now set linguistics aside for a moment and develop the notion of an inheritance network, borrowing some basic definitions from Bob Carpenter [**?**]:[2]

(Inheritance Network) An inheritance network $\mathcal{I}$ is a triple $\langle \mathcal{B}, \sqsubseteq, \# \rangle$ where:

- $\mathcal{B}$ is a finite set of basic elements
- $\sqsubseteq \subseteq \mathcal{B} \times \mathcal{B}$ is the basic *inheritance* relation
- $\# \subseteq \mathcal{B} \times \mathcal{B}$ is the basic *disjointness* relation

(Inheritance/Disjointness) The *inheritance* relation $\sqsubseteq^* \subseteq \mathcal{B} \times \mathcal{B}$ is the smallest such that:

[2] A word about the notational conventions below, which I take to be standard in computer science: The *basic* inheritance and disjointness relations are those that are given explicitly in the formal description of an inheritance network. Inheritance and disjointness *simpliciter*, identified by an asterisk superscript, are the total relations that can be derived from the basic relations.

- $P \sqsubseteq^* P$ (Reflexivity)
- if $P \sqsubseteq Q$ and $Q \sqsubseteq^* R$ then $P \sqsubseteq^* R$ (Transitivity)

The *disjointness* relation $\#^* \subseteq \mathcal{B} \times \mathcal{B}$ is the smallest such that:

- if $P\#Q$ or $Q\#P$ then $P\#^*Q$ (Symmetry)
- if $P \sqsubseteq^* Q$ and $Q\#^*R$ then $P\#^*R$ (Chaining)

The * notation above indicates the *basic* inheritance and disjointness relation, i.e. the relations that must be given explicitly in the definition of $\mathcal{I}$. The total inheritance and disjointness relations for any $\mathcal{I}$ include all of the basic relations, plus all relations that can be derived from the basic relations and the axioms of inheritance and disjointness. In diagramming inheritance networks, I will use solid directed edges to indicate inheritance and dashed undirected edges to indicate disjointness; the absence of an edge will indicate that there is no relation between elements. For example, let $\mathcal{I} = \langle \mathcal{B}, \sqsubseteq, \# \rangle$, where $\mathcal{B} = \{a, b, c, d, e, f\}$, $\sqsubseteq^* = \{\langle a, d\rangle, \langle d, f\rangle, \langle c, e\rangle, \langle b, e\rangle, \langle e, f\rangle\}$, and $\#^* = \{\langle b, c\rangle, \langle d, e\rangle\}$. Figure 1 shows all of the basic relations and some of the derived relations of $\mathcal{I}$.

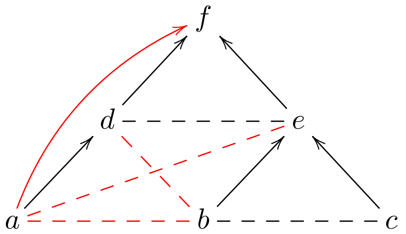

Figure 1

It is given in the definition of $\mathcal{I}$ that $\langle a, d\rangle, \langle d, f\rangle \in \sqsubseteq^*$. Now, $\langle a, f\rangle \notin \sqsubseteq^*$. But since $\langle a, d\rangle, \langle d, f\rangle \in \sqsubseteq^* \vdash \langle a, f\rangle \in \sqsubseteq$ by the transitivity of $\sqsubseteq$, Figure 1 shows $a \sqsubseteq f$. Likewise, $\langle a, b\rangle \notin \#^*$, but Figure 1 shows $a\#b$ because $\langle d, e\rangle \in \#^*, \langle b, e\rangle \in \sqsubseteq^* \vdash$ $\langle a, b\rangle \in \#$ by chaining.

Multiple inheritance is the feature of some systems in which an element c can inherit from more than one element not identical to c, e.g. $\langle c, a\rangle, \langle c, d\rangle \in \sqsubseteq$.

Multiple inheritance is not allowed in all systems, but it will become useful in Chapter 5, so it is allowed here. Figure 2 gives an example of an inheritance network with multiple inheritance:

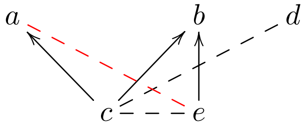

Figure 2

Note that we could not add $a\#b$ to the inheritance network shown in Figure 2, because this would violate chaining. Multiple inheritance is allowed, but restricted. An element c can inherit from multiple parents $a, b, \dots,$ only if none of the parents are disjoint.

The central concept of this framework is:

(Consistent Definition)(cf. [**?**], *Conjunctive Concept*) A set $D \subseteq \mathcal{B}$ is a *consistent definition* on $\mathcal{B}$ iff:

(1) For all $x, y \in D$, it is not the case that $x\#y$.

(2) For all $x \in D$, $y_1, \dots, y_n \in \mathcal{B}$, if $x \sqsubseteq y_i$, then $y_i \in D$ iff there is no $y_j \in D$ such that neither $y_i \sqsubseteq y_j$ nor $y_j \sqsubseteq y_i$.

(3) There exists an α_b (called the *base atom* or *base* of D), such that for all $y_1, \dots, y_n \in \mathcal{B}$, if $y_i \sqsubseteq \alpha_b$, then $y_i \notin D$, i.e. D has a minimal element.[3]

Formally, a consistent definition is a subset of $\mathcal{B}$ containing some atom, called the *base atom* α_b, no disjoint atoms, and a maximal set of atoms $C_{\alpha_b} = \{\beta_1, \dots, \beta_m\}$ such that $\alpha_b \sqsubseteq \beta_1 \sqsubseteq \dots \sqsubseteq \beta_m$ for each α_i. Less formally (and perhaps easier to grasp), we can understand a consistent definition as the set of all atoms from which the base atom inherits along exactly one path in the inheritance network. The following is a non-exhaustive list of consistent definitions shown in Figure 1:

(i) $\{a, d, f\}$

[3] Note that being a minimal element of a consistent definition D does not require being a minimal element of the inheritance network $\mathcal{I}$.

(ii) $\{d, f\}$

(iii) $\{f\}$

(iv) $\{b, e, f\}$

Note that $\{a, d\}$ is not a consistent definition, since it does not contain f, and both a and d inherit from f. Neither is $\{b, e, d, f\}$ a consistent definition, since $e\#d$. $\{b, d, f\}$ is not a consistent definition, since there is a derived disjointness relation between b and d.

And this is an exhaustive list of the consistent definitions shown in Figure 2, which has multiple inheritance:

(i) $\{c, a\}$

(ii) $\{c, b\}$

(iii) $\{e, d\}$

(iv) $\{a\}$

(v) $\{b\}$

(vi) $\{d\}$

Although a and b are not disjoint in Figure 2, $\{c, a, b\}$ is not a consistent definition, since $a, b \in D$ would violate condition (2) for consistent definition. This is the "exactly one path" condition in the "less formal" description of consistent definition given above.

When it is necessary to call attention to particular consistent definitions, they will be given in colors other than black and red, such as in:

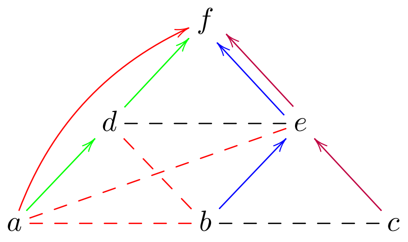

Figure 3

For a somewhat less abstract view of things, lets plug some semantic atoms into Figure 1:

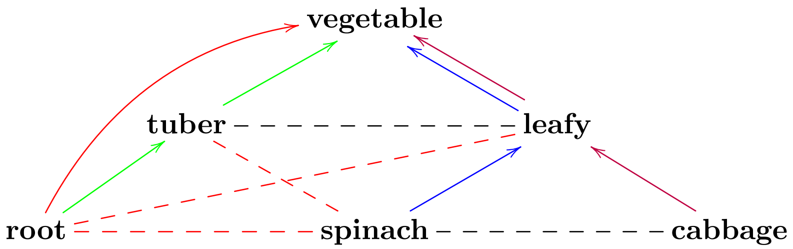

Figure 4

Now, suppose we want a genus-differentia definition for "potato." Assuming potatoes are the only root tuber (a false assumption, but its falsity shouldn't matter for the present purpose), we might take "root tuber" as a definition, where **tuber** serves as a genus and **root** serves as a differentia.[4] This locates "potato" within the inheritance network, from which we can plainly see that it is part of the meaning of "potato" that any potato is a vegetable. Therefore, a full consistent definition on $\mathcal{B}$ of potato is $D_{\text{potato}} = \{\textbf{root}, \textbf{tuber}, \textbf{vegetable}\}$. However, there is no consistent definition $D = \{\textbf{spinach}, \textbf{tuber}, \textbf{vegetable}\}$ because **tuber** and **spinach** are disjoint. Remember Cruse's diagnostic tests for semantic oddness: "It's spinach, therefore it's a tuber" provokes intuitions of oddness, but "It's spinach, therefore it's a leafy vegetable" does not, and so our notion of consistent definition conforms to existing knowledge about semantic content

[4] It might seem strange to call **root** a differetia, since no other subtype of **tuber** is shown in Figure 4; if it's a differentia, it should be differentiating potatoes from something else. This is a fair point, which could easily be remedied by adding $\textbf{stem} \sqsubseteq \textbf{tuber}$ to the model. The fact that it is not shown actually demonstrates an important point about this kind of model: not all concepts/atoms need to be included in every model (in fact, I strongly suspect that it is, in principle, impossible to include all of them). But, this means that, when making use of a particular model, we must remain cognizant of the fact that there are things that are *not* being modeled. This is a general feature of using formal models for analysis, however, not just the models I will present here, and discussion of this topic belongs to a different piece of research.

and captures the semantic relations included in the model at a higher level of abstraction.

(Consistent Definability). We will say that a set S of lexical items is *consistently definable* on $\mathcal{B}$ if and only if there exists a D for every member of S. A lexical item $s \in S$ is said to be *consistently definable* at a point $b \in \mathcal{B}$ if and only if it has a consistent definition D_s containing b. An atom $b \in \mathcal{B}$ is *consistently definable* if and only if there exists a consistent definition D such that $b \in D$. A set of atoms is *consistently definable* if and only if all of its members are consistently definable.

Consistent definitions can be combined into:

(Funtional Role). The *functional role* of an an atom is a function $i : \mathcal{B} \to \mathcal{F})$ such that

$$i(x) = \{D \in \mathcal{F} | x \in D\},$$

where $\mathcal{F}$ is the set of all consistent definitions D on $\mathcal{B}$.

We can take a closer look by considering Figure 5:

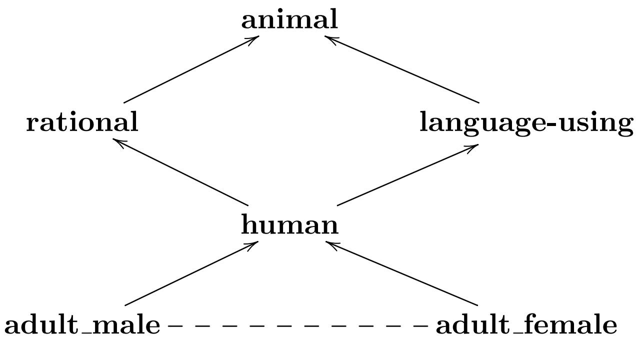

Figure 5

Consistent definitions can be read directly off of the graph:

$$D_{\mathbf{human},1} = \{\mathbf{human}, \mathbf{rational}, \mathbf{animal}\}$$

and

$$D_{\mathbf{human},2} = \{\mathbf{human}, \mathbf{language\text{-}using}, \mathbf{animal}\}$$

are both consistent definitions on $\mathcal{B}$ with $\alpha_b = \mathbf{human}$. Also,

$$D_{\mathbf{man},1} = \{\mathbf{adult\_male}, \mathbf{human}, \mathbf{rational}, \mathbf{animal}\}$$

and

$$D_{\mathbf{man},2} = \{\mathbf{adult\_male}, \mathbf{human}, \mathbf{language\text{-}using}, \mathbf{animal}\}$$

are consistent definitions with $\alpha_{=}\mathbf{adult\_male}$; similarly for $D_{\mathbf{woman},1}$ and $D_{\mathbf{woman},2}$. We can observe that these are all and only the consistent definitions containing **human** as a member. Then:

$$i(\mathbf{human}) = \{D_{\mathbf{human},1}, D_{\mathbf{human},2}, D_{\mathbf{man},1}, D_{\mathbf{man},2}, D_{\mathbf{woman},1}, D_{\mathbf{woman},2}\}.$$

The following theorem, proven in the Appendix, states that any inheritance network over a set $\mathcal{B}$ of consistently definable atoms is isomorphic to the functional role i with domain $\mathcal{B}$. I will first state it in its abstract mathematical formulation and will then explain more concretely what it means in relation to our present purpose.

The genus-differentia inheritance theorem (GDIT):[5] Given $\sqsubseteq$ and $\#$, such that $\sqsubseteq$ is inclusion and $\#$ is disjointness over $\mathcal{B}$, a consistently definable set of atoms, there exists a non-empty functional role i such that $a \sqsubseteq b \Leftrightarrow i(a) \subseteq i(b)$ and $a \# b \Leftrightarrow i(a) \cap i(b) = \varnothing$.

2 Philosophical implications of the GDIT

There are a number of ways of understanding what the GDIT does for us. I take it that these are simply different perspectives on the GDIT, not actually distinct facts that the GDIT entails. What the theorem says is that (a) inheritance networks are isomorphic to the function i. When we state

[5] Thanks to Cody Roux for helping me identify the appropriate mathematical tools for stating and proving this theorem.

things this way, (b) the GDIT can be seen as a representation theorem[6] for inheritance networks. Since, mathematically speaking, inheritance networks are a class of posets, a representation theorem can do a lot of theoretical work for us. One very important aspect of this is that (c) we can also view the GDIT as a soundness and completeness theorem for inheritance networks in the domain of semantic atoms. Given any inheritance network, it is interpretable by some set of consistently definable atoms, and any set of consistently definable atoms can be modeled by an inheritance network. The framework presented here is a formal system—complete with its own well-formedness conditions, axioms, rules of inference, and semantics—and soundness and completeness are important disiderata for any formal systems.

Inheritance networks are a complete formalization of the function i. This is especially appealing for functionalist readers, who will be apt to say that the content of atoms simply is their functional role in the system. For functionalists, then, the GDIT states (d) semantic content is isomorphic to inheritance networks.

There is a very strong naïve intuition that definitions convey word meaning. When we look into meaning more closely, of course, we find out that this view of meaning is inadequate, but we might search for an aspect of meaning that gives rise to the intuition. And the genus-differentia construction of definition can undoubtedly be modeled by inheritance networks. In fact, it can be seen from the above that a genus-differentia definition, once thought to be a full account of word meaning, can be understood as a set of coordinates that locate a word's meaning within an inheritance network that models some aspect(s) of semantic content.

There appears to be a deep relationship between semantic content and abstract order theory, in particular inheritance ordering. Looking at semantic content through the lens of abstract orders, we can see why object-oriented programming is so valuable in computational linguistics: object-oriented languages make heavy use of the inheritance relation. Also, it is

[6] A representation theorem is a theorem stating that every abstract structure exhibiting certain properties is isomorphic to some concrete structure. In our case, inheritance networks are a concrete structure that is isomorphic to the more-abstract functional role.

a perspective with a great enough degree of mathematical precision to facilitate rigorous investigation of linguistic questions, while being abstract and general enough to accomodate a very wide range of specific formal semantic theories. We can understand lexical semantic content *in general*, above and beyond the analysis pursued in specific formal theories of lexical decomposition.[7] In fact, abstract order theory is *so* general that it will afford us a framework for comparing two distinct fields of research: cognitive science and formal lexical semantics.[8] Therefore, this perspective allows us to represent facts about decompositional theories in a way that is especially disposed toward a cognitive interpretation of atoms.

The GDIT applies to consistently definable concepts as well as atoms. We can just as easily replace the word "atom" with "concept" everywhere it appears in the GDIT and its proof and regard the GDIT as a completeness theorem for inheritance networks in the domain of concepts, instead of in the domain of semantic atoms.

Order isomorphisms are transitive. Since inheritance networks are a kind of partial order, this brings us to the real punchline of this thesis:

> *Since conceptual content and lexical semantic content both have properties that are isomorphic to inheritance networks, they have properties that are isomorphic to each other.*

The GDIT hinges on two things: the notion of consistent definition and on the function i. So we should then ask how they apply to concepts. In the domain of semantic atoms, a consistent definition specified the location of an atom within the inheritance lattice; likewise, in the domain of concepts, a consistent definition locates a concept C within the hierarchy, by specifying a set-inclusion path running from C to its most general superordinate. Note

[7] Of course, an inheritance network is a specific formal system, and will carry its own limitations. There may be an even more general formal theory of lexical semantic content, which has yet to be discovered, but this does not threaten the observation that inheritance networks provide an advance in generality over current theories.

[8] Cognitive linguistics is, to a very large degree, a response to formal linguistics. Given this connection between cognitive science and formal semantics, one way of reading this essay is as the beginning of an analysis of the relationship between formal and cognitive linguistics. I will not push such a reading here.

that multiple consistent definitions are possible for the same atom, and the function i maps each atom α to the set of all consistent definitions containing α. In the domain of concepts, i maps each concept C to the set of all set-inclusion paths containing C, i.e. the set of all hierarchy relations in which C participates. Therefore, $i(C)$ is the *functional role* of C in the hierarchy.

The functional role $i(x)$ is a representation of all of the inheritance relations that hold between the semantic content of an atom x and the semantic content of all other atoms in a linguistic theory. Since $\mathbf{adult\_male} \sqsubseteq \mathbf{human}$, by the GDIT we should see that $i(\mathbf{adult\_male}) \subseteq i(\mathbf{human})$. It is left to the reader as an exercise to verify that this is the case. $i(x)$ is also a representation of all of the hierarchy relations that hold between a concept x and all other concepts (included in the model). Since atoms are representations of semantic content and concepts are the providers of semantic content, i is an extremely powerful consolidation of a tremendous amount of information about the semantics of content words.

The analogy between concepts and atoms is quite direct. One *apparent* difference between a conceptual hierarchy and inheritance networks is that, given the way a conceptual hierarchy has been described here, there is no analog of the disjointness relation, which was necessary to model certain semantic entailments that arise out of semantic content. I do think that a disjointness relation can be built into the hierarchy, and it is obvious how theory theories, at least, could accomodate disjointness. Perhaps some of the more extreme versions of the exemplar theory, in which a concept is regarded as a list of *all* objects in the category, could accomodate disjointness as well, but these versions of the exemplar theory are controversial.

If concepts are to do the work required of them here, then they cannot themselves rely on linguistic meaning in any way for their existence or description. We need not be concerned with showing this for *all* concepts. We need only show that this holds for a certain class of concepts, which are viable as an interpretation of atoms. Atoms are supposed to be the most basic components of lexical meaning. It seems right, then, that we might look for some most basic set of concepts. In fact, there is a well-studied set of concepts with exactly this feature.

There is a *basic level of categorization*, which was first studied by Roger Brown [**?**], later followed by Berlin [**?**], Malt [**?**], and others. These researchers have found repeatedly that the most basic level of categorization is neither the most nor the least general level of the hierarchy. Instead, it lies somewhere in the middle. The first precise operational description of the basic level was given by Rosch, et al., in a series of experiments from the 1970s [**?**] [**?**]. Their original description was found to be problematic [**?**], but a number of others have been offered. In particular, Murphy and Brownell have offered a metric in which basicality is determined by *informativeness* and *distinctiveness* [**?**]. Maximizing informativeness, a measure of the amount of information associated with the concept, predicts a basic level that is lower in the hierarchy, while maximizing distinctiveness, a measure of the dissimilarity to other concepts with a common superordinate category or genus, predicts a basic level that is higher in the hierarchy. Informativeness and distinctiveness combine into *differentiation*; maximizing differentiation predicts a basic level somewhere near the middle of the hierarchy. The most robust advantages for a basic level that maximizes differentiation have been found when using artificial, purely perceptual categories that are unfamiliar to test subjects and are not associated with any known word, which is very strong evidence in support of the non-linguistic or pre-linguistic nature of basic concepts [**?**, pg. 220].

Of course, we need atoms at more than just the basic level of hierarchy, and therefore more than just the basic level of concepts needs to be non-linguistc. Unfortunately, the empirical research is not particularly helpful here. Subordinate concepts have not been very well-studied. But subordinate concepts aren't a major concern at any rate. In the inheritance networks of linguistic theories, there is a definite bottom level, which is not the case with concepts. Moreover, the bottom level usually given in examples of lexical decomposition is the kind of thing that is normally seen at the basic level of concepts, e.g. **book**, **dog**. More important are the superordinate concepts, which are obviously an important part of inheritance networks over semantic atoms; we see such general atoms as **event** and **thing**. But research on superordinate concepts is unfortunately linguistic at the outset. Most research into this area is about the linguistic behavior of words that are taken to represent superordinate concepts (such as the fact that superordinate concepts can sometimes be named by mass nouns [**?**]). This research, it must be pointed out, does not establish that superordinate concepts *are* dependent on some linguistic content, but that the question under consideration is itself linguitic: *What sort of linguistic representation will we tend to give for superordinate concepts?*

Since the empirical research in this area is not helpful for the present purpose, I limit myself to giving a plausible description of how superordinates might not depend on linguistic content. Superordinate concepts are representations of the genera for the basic concepts, united by some feature possessed by all members of the genus. The basic concepts united under a superordinate concept form a similarity group, in much the same way that the basic level concepts are similarity groups between objects. Surely most of us have recognized similarity between abstract ideas without being able to put in words the way in which they are similar. This experience lends some credence to the idea that superordinate concepts might be obtainable without reliance on linguistic meaning. Ahn and Medin [**?**] [**?**] have provided a model of concept formation that does not rely on linguistic content; they do not distinguish between basic and superordinate levels in their research, and it appears that the concepts formed by their subjects are basic level concepts (which makes sense, since the basic level concepts are the ones we would expect them to form most readily), but their model is extendable to the superordinate domain, provided similarity does not necessarily appeal to linguistic meaning.[9]

Here, theory theories seem slightly more problematic than either prototype or exemplar theories. For theory theories, concepts are characterized in part through the linguistic relations in which they participate, even at the basic level, since concepts are defined by their role in a broader scientific theory. Exemplar and prototype theories do not face this difficulty. How-

[9] In fact, in the same way that an inheritance ordering naturally falls out of genus-differentia definition, an inheritance ordering will naturally fall out out of Ahn's and Medin's model.

ever, only theory theories can obviously and straightforwardly accomodate disjointness. More work needs to be done in order to resolve this tension, which is impossible to undertake in the present paper. Such tensions between theory theories and prototype/exemplar theories are relatively common in the research on concepts, so it should be unsurprising that a tension arises here. Greg Murphy has observed that these tensions seem to indicate that we should want some sort of hybrid between theory theories and prototype/exemplar theories, but research in this area is not yet advanced enough to give a clear answer as to how such a hybrid will work [**?**].