Concepts, Definitions, and Inheritance

Interpreting the atoms of lexical decomposition

by

DAVID ZORNEK

Theories of Lexical Decomposition

To begin, we will look at some examples of lexical decomposition, highlighting those aspects of each theory that are relevant to the present discussion. Each theory is formulated within its own framework, based on whatever set of foundational assumptions are held by each theorist: Jackendoff models internal representations of meaning, as a total theory of semantics, while Levin and Rappaport Hovav model the semantic features of verbs that have consequences for syntax. James Pustejovsky's *Generative Lexicon* (GL) seeks to improve on traditional lexicons by decomposing words in a way that allows new word senses to be generated, rather than requiring all word senses to be stored in the lexicon as independent entries. We will not here be concerned with these foundational differences; regardless of which approach to lexical decomposition we look at, the general *method* of analysis—the method of composing meaning out of atoms, by means of some structural rules of composition—is the same. Our primary concern here is to understand this general method, rather than the nitty-gritty details of each individual theory.

In particular, we will identify two properties that are shared by all decompositional theories, one essential and one (seemingly) accidental. It is essential to a theory of lexical decomposition that it includes some sort of semantic *atom* in its ontology; decomposition simply *is* breaking semantic content down into atoms. Although not essential to decomposition per se, we will observe that the atoms of all decompositional theories are related to each other by *inheritance*, which will be defined below. The interpretation of atoms offered in this thesis will explain why inheritance appears in decomposition so consistently, turning this accidental property into an essential one.

1 Formal Thoeries of Lexical Decomposition

1.1 Jackendoff's cognitive semantics

The meaning of a lexical item (or a portion of its meaning, at least) is given in a *feature structure* in which types are assigned to different "features" or aspects of the word's meaning. The basic form of one of Jackendoff's feature structures is:

$$\begin{bmatrix} \textbf{event, thing, place,} \ldots \\ \text{token/type} \\ F(\langle \text{Entity}_1, \langle \text{Entity}_2, \langle \text{Entity}_3 \rangle\rangle\rangle) \end{bmatrix}$$

The top feature in the structure structure specifies an *ontological category* under which the lexical item falls. Ontological categories provide a partition of the ways in which our cognitive architecture allows us to represent word meaning. The middle feature is an immediate *inheritance relation* (explained below) in which the word participates (normally by specifying a type from which the given word inherits), and the third and final feature is the argument structure of the word, i.e. the types of arguments that the word can take, which will change depending on the sense in which the word is being used. For example, take the following typed feature strucutures for "novel" and "begin":

$$\textbf{novel} \begin{bmatrix} \textbf{thing} \\ \textbf{book} \\ F(\langle \textbf{property} \rangle) \end{bmatrix}$$

$$\textbf{begin} \begin{bmatrix} \textbf{event} \\ \textbf{transition} \\ F(\langle \textbf{animate\_object}, \langle \textbf{event} \rangle\rangle) \end{bmatrix}$$

"Begin" takes a primary argument of type **animate_object** (since only animate objects can begin actions) and a secondary argument of type **event** (since only events can be begun).[1] "Novel" might take arguments[2] In the case of verbs, the functional arguments turn out to be the grammatical arguments, of type **property**, as it is used in the sentence "This novel is long"; but, in the sentence "The novel is on the table," the argument type is **place**. These are differences between senses of "novel," which are largely determined by the syntactic structure of the sentence and a grammatical conceptual structure inherent in human cognition. The details of how this works are interesting, but not relevant to the present project.

The ontological category and immediate inheritance relation, however, *are* relevant to the present project, so it is useful at this point to explain them.

Jackendoff's types are organized according to the *inheritance relation* $\sqsubseteq$.[3] Inheritance is reflexive and transitive, but not symmetric. Given two atoms α and β such that $\alpha \sqsubseteq \beta$, we say that α inherits from β. Or, α is a *subtype* of β. If there is a third atom γ such that $\beta \sqsubseteq \gamma$, then what the inheritance relation tells us is that $\alpha \sqsubseteq \gamma$ as well. If we let $\mathbf{x}$ be a member of any domain to which the types can apply, we write $\mathbf{x}{:}\alpha$ to indicate that $\mathbf{x}$ is of type α. By inheritance, $\mathbf{x}{:}\alpha \vdash \mathbf{x}{:}\beta \vdash \mathbf{x}{:}\gamma$.

The broadest type, of which all other types are subtypes is the type **entity**. Beneath **entity**, there is a set of *ontological categories*, none of which inherit from each other, but all of which inherit from **entity**. These are **thing**, **event**, **state**, **action**, **place**, **path**, **property**, and **amount**. This sort of system might remind the reader of Aristotle's *Categories*, which

[1]We might quibble over whether these type assignments are the *right* ones for "begin." It's not important here whether I accurately convey the specific types involved in the semantics of "begin," and these will suffice at least to get a handle on how Jackendoff's feature structures are supposed to work, which is the real goal here.

[2]Jackendoff uses the term "argument" here in a non-linguistic sense, i.e. it does not mean "grammatical argument." He does little to explain what other sense of "argument" he is using, but since Jackendoff is a staunch functionalist, and since everything he calls an argument is an input to the function F, which maps properties onto meanings, I read him as using "argument" in the sense of arguments of a function.

[3]Jackendoff does not call the relation inheritance; I have modified his terminology in order to make my own terminology consistent throughout this paper. He calls the relation an "IS-A" relation or a "type-token" relation, both of which are fairly common alternatives.

might be regarded as the earliest example of a type-inheritance theory of lexical semantics. Every lexical item will fall under one of the ontological categories.

At first glance, Ray Jackendoff seems to advocate a version of my own view. In [4], he argues that word meaning is decomposed into concepts, but his usage of the word "concept" is broader than mine. For Jackendoff, a concept is any mental representation, and it includes subjective representations of categories (which is, more or less, what cognitive scientists mean when they talk about concepts), subjective representations of propositions, mental representations of individual objects, and perhaps even the cognitive structural rules which Jackendoff believes to be the source of basic grammatical structure for natural languages.

Jackendoff's notion of concept is too broad to do the work that will be required here. The connection I draw between semantic and conceptual content will rely on empirical results pertaining to the hierarchical organization of concepts (in the cognitive scientist's sense). These results have not been seen for other mental representations.

Moreover, his notion of concept is entirely subjective; that is, Jackendoff's concepts explicitly have no important connection to the external world. This is largely due to his view on the nature of mind, which is little more than a more readable rendition of Kant's *Transcendental Aesthetic* [4, Ch. 2]. Our faculty of sensation is somehow roused into action by external stimuli. Sensations are used as a kind of paint by the mind, which creates a subjective picture or "projected world" (cf. Kant's phenomenal world) by applying the paint in accordance with concepts, which act as a kind of blueprint. It is concepts, not the external world, that determines what we see in the projected world. And the projected world is *all* we see, all our words can possibly refer to. One of our goals here is to explain how words hook up with the external world, but Jackendoff (and Kant) have reinterpreted the word "world" in such a way that it becomes non-sensical to talk about the external world at all.

I have serious reservations about whether Jackendoff should be allowed to reinterpet "world" in this way, but arguing this point belongs to an entirely different project. For the present purpose, it is taken as an explicit assumption that our words do connect with the external world in some way. Under this assumption, we then ask the question "How do words connect to the external world?" Since Jackendoff has explicitly denied that words (or concepts) have any important relation to the external world, making this assumption places us outside the scope of Jackendoff's framework.

Nevertheless, Jackendoff's theory is decompositional and therefore exhibits the same features of other decompositional theories that will be important here. Where decomposition is concerned, it is type "concepts" (in the Jackendoff-ian sense) that are taken as atomic.

As will be seen below, there will be a fundamental connection between these parts of the feature structure and semantic content. Before spelling out this connection explicitly, however, we will look at a few more examples of lexical decomposition.

1.2 Levin and Rappaport Hovav's verb decomposition

In [5], Levin and Rappaport Havov offer a decompositional semantics for verbs specifically. In particular, they model those aspects of verb semantics that have consequences for which grammatical arguments must be, might be, or cannot be syntactically realized in a sentence. Like Jackendoff, they hold the view that there is some definite set of ontological categories that exist as part of the semantic theory, i.e. there is some universal set of atoms from which all verb meaning is constructed. In fact, they accept Jackendoff's ontological categories, *plus* an additional set of fixed ontological categories that inherit from Jackendoff's **event** type. Among these additional ontological categories are **cause**, **become**, and **act**; in another paper [6], they give a long list of others that are commonly seen in theories of verb decomposition. The subtypes of Jackendoff's other ontological categories (which are not dubbed as "ontological categories"), are left explicitly open-ended; that is, they can be whatever and however many in number are required to give a semantic analysis of all verbs in a natural language.

Some of the ontological categories in this additional set are called *pred-*

icates, and they will determine the structure of the semantic entry, while the others are called *constants*, which provide semantic content and act as arguments for the predicates.

We cannot give a semantic entry for "novel" in their system, since it deals only with verbs, but we can give an entry for "begin":

$$[[x\ \mathbf{act}]\mathbf{cause}[x\langle ACTION\rangle]],$$

where $ACTION$ is a variable standing for any subtype of the constant **action**. If some agent x *begins* to perform some *action*, then x *acts* to *cause* x to be in a state of *action.* For example, the event structure for the event described by (2) "Maria began reading" is

$$[[\text{Maria}\ \mathbf{act}]\mathbf{cause}[\text{Maria}\langle\mathbf{reading}\rangle]].$$

Again, we have a set of atoms that provide semantic content, i.e. the constants. And again the use of these atoms relies on an inheritance relation: inheritance from **action** is a constraint placed on the content of arguments for "begin."[4]

Levin and Rappaport Hovav's semantic theory is not merely an abstract exercise. Computational lexical resources exist which have real-world technological implementations such as automated translation, corpus annotation, treebanking, automated sentence parsing. These implementations are useful for application in artificial intelligence, speech recognition, language learning software, etc.

VerbNet, a lexical resource created and managed by Karin Kipper-Schuler, Martha Palmer, and others at University of Colorado, is currently the largest on-line verb lexicon for English. Lexical entries are organized into verb classes based on the predicates of Levin and Rappaport Havov[5], and it is

[4]Section 3.1 of this chapter includes some more examples of this kind of constraint. All three examples in this section are in the Levin class of *change of state* verbs, which share the structural template $[\mathbf{become}[y\langle RES\text{-}STATE\rangle]]$. Any constant that can serve as a value for the variable $RES\text{-}STATE$ must be a subtype of the constant or **resulting_state**.

[5]In VerbNet, the predicates are called *thematic roles*, following [3], but they come down to the same thing as Levin and Rappaport Havov's predicates. I call them predicates

perhaps in the implementation that we can most-readily observe the type-inheritance system of Levin and Rappaport Hovav's content-bearing constants (or atoms).

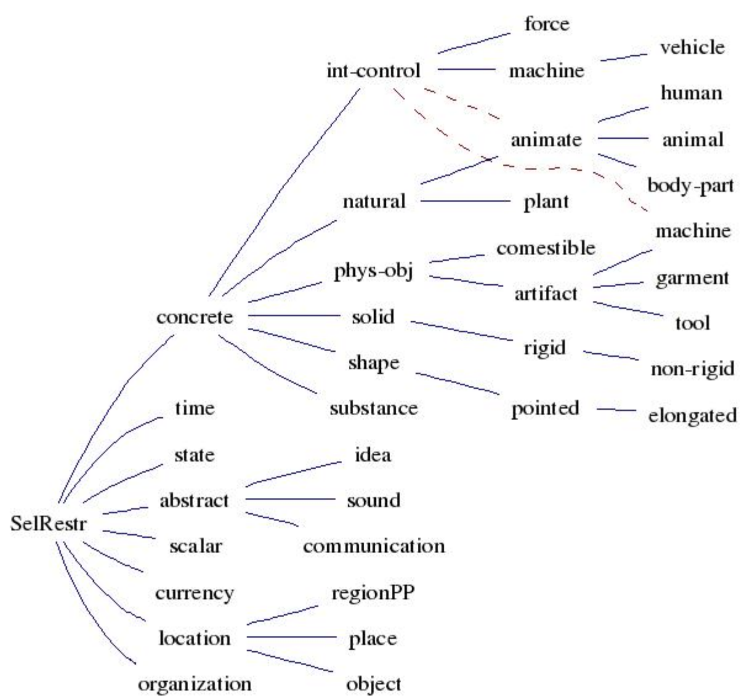


Figure 1: A sample type inheritance hierarchy from VerbNet

2 Moral of the story: Lexical Semantic Content and Inheritance

2.1 Lexical Semantic Content

In any decompositional theory, there is a structural component and a content-providing component. Levin and Rappaport Havov [6] have made this exact

because I have already introduced this term and VerbNet is explicitly based on their theory.

point for a particular class of decompositional theories, called Lexical Conceptual Structures (LCS). Although LCSs are theories of verb decomposition, their comments apply to lexical decomposition in general. Many words might have the same general structure for Levin and Rappaport Hovav's lexical entries, only differing in their semantic content, which is represented in the form of semantic atoms. For example, given the structural template $[\mathbf{become}[y\langle \mathit{RES\text{-}STATE}\rangle]]$, where *RES-STATE* is a variable ranging over the domain of resulting states, we have the following three lexical entries:[6]

1. *dry*: $[\mathbf{become}[y\langle \mathbf{dry}\rangle]]$
2. *widen*: $[\mathbf{become}[y\langle \mathbf{wide}\rangle]]$
3. *dim*: $[\mathbf{become}[y\langle \mathbf{dim}\rangle]]$

It is obvious that "dry," "widen," and "dim" all have very different meanings, yet the structure of these lexical entries is the same, different only in the atom that occurs in the argument for **become**. The part of each lexical entry that is idiosyncratic to each word is the content provided by the atom. The same point applies to Pustejovsky and Jackendoff: their formal representations all have the same basic structure, so differences in semantics between words can only arise through the types that are assigned within each entry in the lexicon.

In lexical decomposition, semantic content is always inherited from some basic atom. But this is only possible if atoms have content, so it makes sense to inquire after the content of atoms.

David Lewis [7] has given a compelling argument in the domain of sentential decomposition that atoms must be given an interpretation in order for a semantic theory to do any of the work we require from such a theory. A decompositional theory provides a set of atoms or "markers," which are essentially a lexicon for the theory. The compositional rules provide a syntax according to which we combine the atoms into some sort of structure representing the semantics of whatever linguistic unit is in the domain

[6] "Lexical entry" is a general term that applies to any structure or item that is used to convey the meaning of a word in the lexicon of a semantic theory.

of the theory (in Lewis's case, the linguistic units under consideration are sentences; in ours, they are words). "But," Lewis writes,

> we can know the Markerese translation of an English sentence without knowing the first thing about the meaning of the English sentence: namely, the conditions under which it would be true. Semantics with no treatment of truth conditions is not semantics. Translation into Markerese is at best a substitute for real semantics, relying either on our tacit competence (at some future date) as speakers of Markerese or on our ability to do real semantics at least for the one language Markerese.

Proponents of decompositional theories are able to get by without providing content to their atoms precisely because they and their audience are speakers of some natural language. The atoms of lexical decomposition are, *invariably*, represented by the same signs as words in natural language. Since we are all able to understand the words represented by these signs, in virtue of the fact that we are all fluent speakers of that natural language, we are able to make sense of decompositional lexical entries even without a "real semantics" for English or Markerese. Theorists of lexical decomposition rely on fluency to supply content to atoms where none is given within the theory. One of the main goals of this project is precisely to indicate one way in which we can do "real semantics" for Markerese, at the same time situating our decompositional theories within the broader context of "real semantics" for natural language.

If we follow Lewis in thinking that uninterpreted atoms do not have content, but want to retain the decompositional approach, then we will want to compose word meaning out of some atom whose content we understand independently of the decompositional theory. Absent some interpretation that provides content to atoms from without, it is difficult to see how we might obtain such understanding. Any theory of lexical decomposition is necessarily unable to non-circularly provide content to its atoms. Within the theory, the meanings of atoms can only be decomposed into other atoms; theories of lexical decomposition are *only* theories of lexical decomposition.

But if the meanings of atoms are given decompositionally, then either: (a) they aren't *actually* atoms of the theory; (b) we will ultimately bottom out with the meanings of some atoms given in terms of themselves; or (c) we have a "decompositional circle," e.g. α is defined in terms of β and γ, β is defined in terms of δ and ϕ, and so on, until we reach some atom defined in terms of α. In any case, we cannot form a coherent view of how a decompositional theory can provide meaning to its own atoms. The current external source of content for atoms is their identification with English words, which retrieve content from fluency of native speakers. But if this is the actual source of content for the atoms, then we have an obvious circularity: linguistic items supply content to atoms, which supply content to linguistic items.

Nevertheless, linguists and philosophers alike (many of them at least) seem to agree that decompositional theories play a crucial role in semantics as a whole.[7] It is not the case that lexical decompositional theories are completely useless until and unless we can find ourselves in possession of a convincing account of the content of its atoms. Any lexical decompositional theory comprises only part of a total science of lexical semantics. Decompositional theories can tell us a great deal about the structural component of meaning.

However, we take ourselves to be talking *about* something when we use words, and except when what we are actually talking about is atoms, to say that atoms are the providers of the content component of meaning doesn't tell us what our words are *about*. When I say, "The sky is clear today," I've said nothing at all about atoms. I've said something about the sky. Whatever sort of thing meaning is, *aboutness* is a major component of it. We might say that "The sky is clear today" is about the sky because the word "sky" refers to the sky. This is on the right track, but some words have meaning without referring to anything, e.g. unicorn, goblin, etc., so

[7]In the domain of sentential semantics, it is wholly uncontroversial that decomposition is the right approach. There is some controversy over lexical decomposition, and die-hard opponents of lexical decomposition may reject what is said here. Justifying lexical decomposition is outside the scope of this paper; I seek to fortify lexical decomposition, given that we already agree on the appropriateness of the decompositional approach to word meaning.

reference cannot be a necessary condition of meaning. As a guiding intuition, we might say: When a word has reference, its reference plays an important role in meaning. In cases where a word has no reference, some story will need to be told about the meanings of words, but I think that what I will argue is compatible with any plausible story that might be offered.

My proposal is that we can find some cognitive representation to interpret atoms in order to provide a non-circular external source of meaning. It seems to me to be intuitively plausible that cognition and meaning go hand-in-hand. Understanding a language involves having a certain cognitive relation to well-formed sentences of that language. But we can say a bit more. Our sense of *aboutness* is at least in part a mental phenomenon, and therefore we should look inward for an explanation of this sense. Moreover, by identifying atoms with some cognitive representation, we might be able to account for *aboutness* while simultaneously screening ourselves off from reference in a way that will leave room for meaning without reference where necessary. This picture should be able to satisfy referentialists such as Lewis, by allowing our cognitive structures to mediate between words and reference while also being palatable to subjectivists such as Jackendoff, who wish to give a theory of internal semantics in which linguistic units have no meaning beyond what is present in the mind.

Two problems with the current situation have already been mentioned: there is no coherent, non-circular way for decompositional theories to provide content to their own atoms without relying on prior understanding of the meanings of words; without content for the atoms, we are unable to account for *aboutness*. Also, we should keep in mind that atoms are supposed to be carriers of semantic content; therefore, any viable interpretation must exhibit some behavior that reflects fundamental observations about semantic content. At this point, we are able to identify three criteria that a successful interpretation of the atoms must meet. I will take the current project to have been successful when I have established an interpretation that

(i) accounts for *aboutness*,

(ii) is not itself dependent on linguistic meaning, and

(iii) exhibits some fundamental property that parallels semantic content. In particular, we want an interpretation that explains why every theory of lexical decomposition involves:

2.2 Inheritance

Inheritance has shown up in every instance of lexical decomposition covered here. In fact, I am aware of no instance of lexical decomposition—even outside of what has been included above—that does not involve an inheritance order over the atoms. The existence of atoms is essential to lexical decomposition; decomposition, after all, is decomposition into atoms. But inheritance seems to show up mostly by accident. There is nothing about decomposition *per se* that necessitates inheritance. We can at least conceive of decomposing word meanings into atoms that are not related by inheritance. This seems to point toward the hypothesis that there is something fundamental about semantic content that exhibits inheritance behavior, apart from the role that semantic content plays in lexical decomposition. In fact, as will be seen in Chapter 3, there is a kind of cognitive representation that exhibits inheritance behavior and which is intuitively plausible as a provider of lexical semantic content—namely, concepts.

3 James Pustejovsky's Generative Lexicon

James Pustejovsky's theory of lexical decomposition, the Generative Lexicon (GL) [9] will provide a case study for developing the conceptual interpretation of semantic atoms, so it will be useful to look at GL in some level of detail.

4 Sense Enumeration Lexicons

GL is a theory of lexical semantics based on LKB [10], which is in turn based on Bob Carpenter's logic of typed feature structures [1]. GL is a model of word meaning in which the mechanisms underlying systematic polysemy allow new word senses to be generated "on the fly," which are not given explicitly in the lexicon. Historically, the simplest and most direct means of handling polysemy has been to allow the same word to be listed multiple times in the lexicon, with each listing storing a different semantics for the word. A precise characterization of this sort of *Sense Enumeration Lexicon* SEL is

> A lexicon L is a *Sense Enumeration Lexicon* if and only if for every word w in L, having multiplie senses $s_1, \ldots, s_n$ associated with that word, then the lexical entries expressing these senses are stored as $\{w_{s_1}, \ldots, w_{s_n}\}$. [9, p. 34]

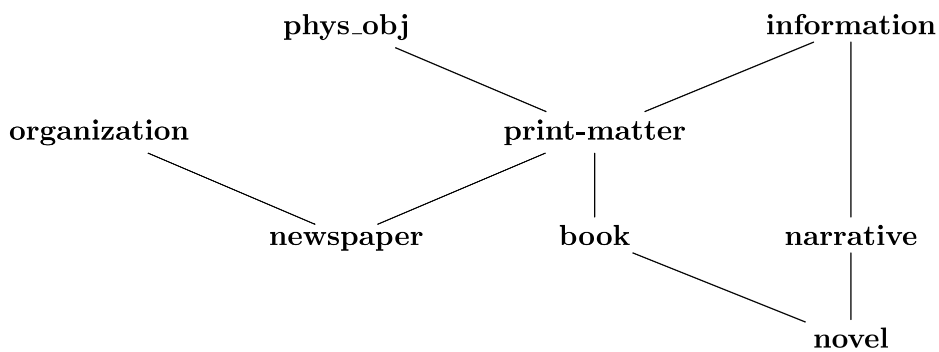
The inadequacies of SELs are spelled out detail by Pustejoveky [9, Ch. 4], and for a fuller discussion of them, the reader is referred to his book. They are not important for our purposes here. What is important is that Pustejovsky introduces the generative mechanisms of GL as a way of avoiding the necessity of enumerating every word sense in the lexicon.

5 Typed Feature Structures

In the GL, the semantics of a lexical item α is represented as a quadruple $\langle \mathcal{A}, \mathcal{E}, \mathcal{Q}, \mathcal{I} \rangle$. Each component is discussed below. The inheritance structure $\mathcal{I}$ is covered first, since it is the main one we are concerned with here. I have included discussion of $\mathcal{A}$, $\mathcal{E}$, and $\mathcal{Q}$ for completeness, but they are not essential to understanding what follows.

5.1 The lexical inheritance structure $\mathcal{I}$

The lexical inheritance structure $\mathcal{I}$ is a lattice which defines what is to be considered as a type and the relations between types. Following Carpenter's Logic of Type Feature Structures, the fundamental relation between types is inheritance. The inheritance lattice is a partial ordering $\sqsubseteq$ over types and we say that α *inherits from* β just in case $\alpha \sqsubseteq \beta$. If α inherits from β, then α is a *subtype* of β, i.e. for any object $\mathbf{x}{:}\alpha$, it is also the case that $\mathbf{x}{:}\beta$. The entire logic of typed feature structures is a spelling-out of the consequences of the inheritance relation. We do not need to be too concerned with the details of the logic here and can simply refer to Carpenter's work where necessary. A sample inheritance lattice is given below:



Neither Pustejovsky nor Carpenter commits us to any specific inheritance structure, nor even a specific set of types. Carpenter, in particular, is explicit that, so far as his logic is concerned, any set of types and any inheritance structure will do, provided they possess a certain set of very general characteristics. We need not be too concerned with Carpenter's conditions; it will be difficult to cook up any set of types or inheritance relations which do not meet them, but which are also plausible for use in the GL. A more important issue arises from the fact that Pustejovsky does not consider the question of whether there are some inheritance structures that are better than others for modeling word meaning. Indeed, Pustejovsky doesn't say much about $\mathcal{I}$ at all, other than what it is. In Chapter 4, a proof will be given that there is an important relation between the hierarchical organization of concepts and inheritance structures and that, by giving an appropriate model of concepts which is sufficient to explain hierarchy, we can get greater insight into $\mathcal{I}$. Importantly, it is my hope that we will get some answer to the question of whether any $\mathcal{I}$ at all will do, or whether only some $\mathcal{I}$s are viable candidates as part of a linguistic structure in the GL.

5.2 The argument structure $\mathcal{A}$

Argument structure is the best-understood of the components and is regarded as a minimal specification of lexical semantics, although it is far from adequate as a complete characterization of the semantics of any lexical item. There are four main types of argument discussed by Pustejovsky:

The grammatical arguments of the lexical item "build" are encoded into a list structure ARGSTR in the following manner:

$$\begin{bmatrix} \textbf{build} & \\ \text{ARGSTR} = & \begin{bmatrix} \text{ARG}_1 = \textbf{animate\_individual} \\ \text{ARG}_2 = \textbf{artifact} \\ \text{ARG}_3 = \textbf{material} \end{bmatrix} \\ \ldots & \end{bmatrix}$$

The indexing of features in the argument structure corresponds to a sort of semantic proximity to the word whose semantics are given in the feature structure. A verb's subject is the most proximal, while the direct object is the next most proximal, and indirect object even less proximal, etc. Indexing is negatively correlated with proximity. For a verb, the argument structure will give an account of the grammatical arguments expected by the verb, while for a non-verb, the argument structure will correspond, roughly, to the words behavior when used as the argument for a verb. This is best understood through example; Chapter 5 will provide a number of examples that illustrate the argument behavior of non-verbs.[8]

5.3 The extended event structure $\mathcal{E}$

The extended event structure will not be important for the purposes of this thesis and is included merely for completeness and to prevent the reader from feeling overwhelmed by their presence in the typed feature structures

[8] Like Jackendoff, Pustejovsky's use of the term "argument" is somehwat misleading for linguists. Pustejovsky is, fundamentally, a functionalist (in fact, his types are explicitly functions), and I read him as meaning "argument of a function." This reading is corroborated by the fact that Pustejovsky's formal system is explicitly based on Ann Copestake's LKB [2], whose argument structures are explicitly explained as inputs to functions.

that will be used in Chapter 5. The following is an example of the event structure for the word “build”:

$$\left[\begin{array}{ll} \textbf{build} & \\ \text{EVENTSTR} = & \left[\begin{array}{l} \text{E}_1 = \textbf{process} \\ \text{E}_2 = \textbf{state} \\ \text{RESTR} =<_\alpha \end{array}\right] \\ \ldots & \end{array}\right]$$

The E_i are features that correspond to the stages of an event that extends through time. In the above example, the act of building something involves a *process* (of building), and it ends with an object being in a *state* of having been built. The feature RESTR defines the order that is placed over stages of an event, e.g. total ordering, partial ording with simultaneity, partial ordering without simultaneity. $<_\alpha$ is Pustejovsky’s symbol for a total order (without simultaneity), which is perfected or completed with the final stage of the event; building ends when the thing we are building is in a state of having been built, and the act of building and the state of having been built cannot occur simultaneously.

5.4 The qualia structure $\mathcal{Q}$

Pustejovsky’s qualia structure is borrowed from Aristotle’s four modes of explanation or four causes. The analysis given here will not depend on a detailed understanding of what they qualia are; we need only understand that they provide semantic information that should be accessible by native speakers of a language when they use a word. In fact, generative lexicons can be created in which the generative mechanisms depend on features other than qualia structure, so there is nothing essential about Pustejovsky’s choice of qualia. His choice of the qualia is largely predicated on Moravcsik [8], who argues that Aristotle’s four causes serve as a generative mechanism for creating new word senses. The qualia structure QUALIA is constituted by assigning types to the four following features:

1. *Constitutive.* The material properties of an object; the relation be-

tween an object and its proper parts. Some examples are:

(a) Material

(b) Weight

(c) Parts and component elements

2. *Formal.* The property according to which an object of a particular species is regarded also as a member of some broader genus. Some examples are:

    (a) Orientation

    (b) Magnitude

    (c) Shape

    (d) Dimensionality

    (e) Color

    (f) Position

3. *Telic.* The purpose or function of an object. Some examples are:

    (a) Purpose for which an object was created by some agent.

    (b) A built-in function or aim toward which the natural activities of an object point.

4. *Agentive.* The origin of an object or factors involved in its "bringing about." Some examples are:

    (a) Creator

    (b) Artifact

    (c) Natural kind[9]

    (d) Causal Chain

[9]Pustejovsky lists this, but I am hesitant to do the same, due to some confusions and/or skepticism I have pertaining to natural kinds, which are off-topic from the present discussion.

As an example, consider the following QUALIA structure for "novel":

$$\begin{bmatrix} \textbf{novel} & \\ \ldots & \\ \text{QUALIA} = & \begin{bmatrix} \text{CONST} = \textbf{narrative} \\ \text{FORMAL} = \textbf{book} \\ \text{TELIC} = \textbf{reading} \\ \text{AGENT} = \textbf{writing} \end{bmatrix} \end{bmatrix}$$

This simple listing of qualia values "tells us nothing about how a particular lexical item denotes, however. For example, although a novel's purpose is the activity of reading and it comes about by someone writing it, we do not want to claim that the comon noun *novel* actually denotes such activities." The qualia provide information *about* what is denoted by the word.

6 Type Coercion

The main mechanism of polysemy for nominals is *type coercion.* As we have seen, the typed feature structure for verbs will demand that the verb take arguments of a specific type, e.g. "build" takes a subject of type **animate_individual** and an object of type **artifact**. However, we can unproblematically supply verbs with arguments that differ from their designated types, by means of a generative rule that *coerces* a word's argument behavior to the type expected by a verb. Type coercion will be explained more fully in Chapter 5, where it will become necessary to understand how the formal system of Chapter 4 can be applied to answer some specific questions about concepts and theories of lexical decomposition.

References

[1] Bob Carpenter. *The Logic of Typed Feature Structures.* Number 32 in Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, Cambridge, 1992.

[2] Ann Copestake. *Implementing Typed Feature Structure Grammars.* CSLI Publications, Stanford, 2002.

[3] R. Jackendoff. *Semantic interpretation in generative grammar.* MIT Press, Cambridge, 1972.

[4] Ray Jackendoff. *Semantics and Cognition.* The MIT Press, Cambridge, MA, 1983.

[5] B. Levin and M. Rappaport Havov. Building verb meanings. In M. Butt and W. Geuder, editors, *The Projection of Arguments: Lexical and Compositional Factors.* CSLI Publications, Stanford, CA, 1998.

[6] B. Levin and M. Rappaport Havov. Lexical conceptual structure. In K. von Heusinger, C. Maienborn, and P. Portner, editors, *Semantics: An International Handbook of Natural Language Meaning.* Mouton de Gruyter, Berlin, 2008.

[7] David Lewis. General semantics. *Synthese*, 22(1):18—67, 1970.

[8] J.M. Moravcsik. Aitia as generative factor in Aristotle's philosophy. *Journal of Philosophy*, 14:622–636, 1975.

[9] James Pustejovsky. *The Generative Lexicon.* The MIT Press, 1998.

[10] Antonio Sanfillipo. LKB encoding of lexical knowledge. In Ted Briscoe, Valeria De Paiva, and Ann Copestake, editors, *Inheritance, Defaults, and the Lexicon*, Studies in Natural Language Processing, pages 190—222. Cambridge University Press, Cambridge, 1993.