

## 8. Basic Processing in Ontological Semantic Text Analysis

Text analysis is only one of the processes supported by ontological semantics, albeit a central one. An ontology replete with the representations of complex events and objects can also support reasoning in such applications as information extraction, question answering and advice giving. The various applications differ in the measure in which text should be processed in them, from full coverage in MT to spot coverage in IE and summarization to, possibly, no text analysis in planning applications.

The proclaimed goal of ontological semantics as applied to text analysis is to input a text and output a formal expression which is declared to be its meaning representation. This process requires many diverse knowledge sources. This means that building ontological-semantic analyzers and generators may take longer than NLP applications that use other methods. One must bear in mind, however, that no task that requires generation of representations can completely bypass the need for compiling extensive knowledge sources. It is only when no representations are sought can modules even consider relying on purely corpus-based statistical methods as the backbone of processing (cf. Section 2.6.2.1 above).

In this chapter, we present the process of text analysis in ontological semantics. We will remain at the conceptual level and will not go into the details and issues related to potential or actual implementations of these processes. In other words, for each process we will describe the task that it performs, specify its input and output data and the requirements this processing module imposes on static knowledge sources in the system, such as lexicons or the ontology. We will pay special attention to the issue of potential failure of every processing module and ways of recovering from such failures.

### 8.1 Preprocessing

While ontological semantics concentrates on the issues of meaning, no serious NLP application can afford to avoid dealing with non-semantic processing, such as morphology or syntax. The output of these modules provides input and background knowledge to the semantic modules of any ontological-semantic application.

#### 8.1.1 Tokenization and Morphological Analysis

Input text comes in many guises: as plain text or as text with some kind of mark-up, such as SGML, HTML or XML (see Figures 1 and 2). Text, for instance, some newspaper headlines, may come in all-caps. Some languages, for instance Chinese or Hebrew, do not have the distinction between capital and lower-case letters. Languages vary in their use of punctuation: for example, Spanish uses inverted exclamation marks at the beginning of exclamatory sentences. Languages use different means of rendering dates, e.g., May 13, 2000 and the thirteenth of May, 2000 and numbers (for example, Japanese breaks numbers into units of 10,000 and not 1,000, as most European languages) as well as acronyms, e.g., *pur* for *pursuit*, *Mr.* for *Mister* and *UN* for *United Nations*). An NLP system must recognize all this material and present it in a standardized textual form. The module responsible for this functionality is often called the tokenizer.

```

<!-- Yahoo TimeStamp: 950652115 -->
<b>Tuesday February 15 5:01 PM ET</b>
<title>'American Beauty' Leads Oscar Nods</title>
<h2>'American Beauty' Leads Oscar Nods</h2>
<!-- TextStart -->

<p><font size ="-1"><i>By DAVID GERMAIN AP Entertainment Writer </i></font><p>BEVERLY HILLS, Calif. (AP) - The Oscars embraced dysfunction and darkness Tuesday, bestowing a leading eight nominations on the suburban burlesque " <a href="http://movies.yahoo.co/ shop?d=hv&cf=info&id=1800018623">American Beauty</a>" and honoring movies about abortion, death row and the tormented souls of the dead.<p>The top nominees included " <a href="http://movies.yahoo.com/ shop?d=hv&cf=info&id=1800025331">The Cider House Rules</a>" set in a combination orphanage and abortion mill; " <a href="http://movies.yahoo.com/shop?d=hv&cf=info&id=1800019665">The Sixth Sense</a>" about a boy from a broken home who can see ghosts; and " <a href="http://movies.yahoo.com/shop?d=hv&cf=info&id=1800025551">The Green Mile</a>" about the bonds between prison guards and condemned men.<p>Those four movies, along with " <a href="http://movies.yahoo.com/ shop?d=hv&cf=info&id=1800025632">The Insider</a>" a film about a tobacco industry whistle-blower, were nominated for best picture.<p>The top acting categories also were heavy on family dysfunction.<p>The best-actor candidates included <a href="http://search.yahoo.com/bin/search?p=Kevin%20Spacey">Kevin Spacey</a> in "American Beauty" as a dad who blackmails his boss, smokes pot with a son kid and flirts with his daughter's cheerleading friend.
<!-- TextEnd -->

```

**Figure 30. A document with HTML encodings. Only a part of this material must be processed by an NLP system.**

The next stage in preprocessing is morphological analysis of the results of tokenization. A morphological analyzer accepts a string of word forms as input and for each word form outputs a record containing its citation form and a set of morphological features and their values that correspond to the word form from the text. (A number of detailed descriptions of approaches to morphological analysis exist, e.g., Sproat 1992, Koskenniemi 1983, Sheremetyeva *et al.* 1998, Megerdooomian 2000).

Both the tokenizer and the morphological analyzer rely on static resources:

- “ecological” rules for each language that support tokenization (one example of this would be to understand a sequence of a number followed by a period (.) in German as an ordinal numeral, e.g., 2. means *zweite*, ‘second’) and
- morphological declension and conjugation paradigms (e.g., all the possible sets of forms of French verbs, indexed by the value of corresponding features), morphophonological information about stem alternations, and other types of knowledge needed to produce a record, such as {“*vendre*, Past Indefinite, Third Person, Singular”} from the word form *vendait*.

As usual in NLP, there is no guarantee that the static knowledge sources are complete—in fact, one can practically guarantee that they will be incomplete at any given time! In addition, the set of processing rules can contain omissions, ambiguities and errors. For example, the German tokenization rule above will fail when the symbol 2 is put at the end of a sentence. The rule will tokenize the input sequence 2. as <number-ordinal second> while the correct tokenization may, in fact, be <number: 2> <punctuation: period>.

'American Beauty' Leads Oscar Nods

By DAVID GERMAIN AP Entertainment Writer

BEVERLY HILLS, Calif. (AP) - The Oscars embraced dysfunction and darkness Tuesday, bestowing a leading eight nominations on the suburban burlesque "American Beauty" and honoring movies about abortion, death row and the tormented souls of the dead.

The top nominees included "The Cider House Rules" set in a combination orphanage and abortion mill; "The Sixth Sense" about a boy from a broken home who can see ghosts; and "The Green Mile" about the bonds between prison guards and condemned men.

Those four movies, along with "The Insider" a film about a tobacco industry whistle-blower, were nominated for best picture. The top acting categories also were heavy on family dysfunction.

The best-actor candidates included Kevin Spacey in "American Beauty" as a dad who blackmails his boss, smokes pot with a neighbor kid and flirts with his daughter's cheerleading friend.

**Figure 31. The text from Figure 1 that will undergo NLP.**

Morphological analyzers often produce ambiguous results that cannot be disambiguated without some further, syntactic or even semantic processing. For instance, if the English string *books* is input to a morphological analyzer, it will, correctly, produce at least the following two variants: "book, Noun, Plural" and "book, Verb, Present, Third Person, Singular." Of course, there will also be errors due to the incompleteness of static knowledge; processing rules can be insufficiently general or, on the contrary, too generalizing. While unknown words (that is, words not in the system's lexicon) can be processed by some morphological analyzers, there is no protection against spelling errors (unless an interactive spell-checker is integrated in the system). This situation will bring additional problems at the lexical look-up stage.

Improvement of tokenization and morphological analysis is obtained through manual correction of the static resources as well as through integration of additional tools, such as spell-checkers and methods for treating unexpected input (see Section 8.4.3 below).

### 8.1.2 Lexical Look-up

Once the morphological analyzer has generated the citation forms for word forms in a text, the system can look them up in its lexicons, including the onomasticon for names, and thus activate the relevant lexical entries. These lexical entries contain, as the reader knows by now, a variety of types of information, including information concerning syntax and lexical semantics, but also morphology. The latter information is used to double-check and, if necessary, help to disambiguate the results of morphological analysis.

Lexical look-up can produce wrong results for misspellings and fail to produce results for both misspellings and *bona fide* words that are not in the lexicon. For example, many English texts may have Latin (*exeunt*), French (*frisson*), German (*Angst*), Spanish (*paella*), Italian (*dolce*), Russian (*perestroika*) and other words, not all of which would have been accepted as English words and therefore listed in a typical English lexicon. Still more problematic are proper names. Even if

large onomasticons are built, one can be virtually certain that many names will not have been collected there. In some cases, there will be ambiguity between proper and common readings of words—the name *Faith*, when starting a sentence, will be ambiguous with the common reading of the word. Additional difficulties will be brought about by multi-word lexical units, both set phrases (*there is*) and words with discontinuous components (*look up*). Conversely, some words are compound, spelled as a single word (*legroom*, *spatiotemporal*) or hyphenated (*well-known*). One cannot expect to find all of them in the lexicon. Indeed, listing all such entities in the lexicon may not be an option. If a word is not attested in the lexicon, one recovery procedure is to hypothesize that it is compound and to attempt to look up its components. However, at least for Swedish, as reported by Dura (1998), serious difficulties exist for automatic resolution of compounds, which means that the lexicons for Swedish (and, most probably, other compounding languages, such as Hungarian) will have to include the compounds, thus complicating lexical acquisition.

There are two basic ways of dealing with the failures of lexical look-up. First, a system may insist that all unknown words be checked manually and either corrected (if they were misspelled) or added to the lexicon. Second, a system of dealing with unknown words can be built that not only processes compounds but also carries out all the processing that is possible without knowing the stem or root of the word in question (for instance, guessing morphological features, often the part of speech and other syntactic properties but never the meaning). For example, in the Mikrokosmos implementation of ontological semantics, unknown words are treated very casually—all of them are assigned the part of speech Noun and their meaning is constrained trivially: they are declared to be children of the root ontological concept ALL, which amounts to saying that they carry no meaningful semantic constraints. Remarkably, even this simplistic treatment helps (see Beale *et al.* 1997 for details).

### 8.1.3 Syntactic Analysis

The task of syntactic analysis in ontological semantics is, essentially, to determine clause-level dependency structures for an input text and assign syntactic valency values to clause constituents (that is, establish subjects, direct objects, obliques and adjuncts). As it is expected that the results of syntactic analysis within ontological semantics never constitute the final stage of text processing, it has an auxiliary status in the approach. One corollary is that in ontological semantics work on optimization of syntactic parsing is not a high priority.

At the same time, ontological semantics does not dismiss syntactic knowledge out of hand, as was done in early computational semantics (cf. Schank 1975, Wilks 1972). While those authors chose to concentrate on proving how far “pure” semantic approaches can take text analysis, we believe in basing text analysis on as much knowledge as is possible to obtain, from whatever source. Syntactic analysis is supported by a syntactic grammar and syntax-related zones (SYN-STRUC—see Section 7.2 above) of the lexicon entries. In addition, the ontological-semantic lexicon supports syntax-to-semantic linking, mapping between syntactic valencies and semantic case roles.

Just as with other modules (and other syntactic analysis systems), syntactic analysis can fail due to a lack of complete coverage or errors in either grammar or lexicon or both, resulting in inappropriate input sentence chunking, incorrect syntactic dependencies, mislabeling of constituent heads and outright failures to produce output. An additional reason for this latter eventuality may be the ill-formedness of an input text.

Besides, realistic grammars are typically ambiguous, which leads to production of multiple syntactic readings, in which case a special ranking function must be designed and built for selecting the best candidate. Fortunately, within ontological semantics, one can defer this process till after semantic analysis is under way—in many cases semantic constraints will impose enough preferences for the correct choice to be made. At no time will there be a declared goal of selecting the “most appropriate” syntactic reading for its own sake. The ultimate goal of ontological semantics is producing text meaning representations using all available means. Semantic resources and processors are, naturally, a central component of such a system, and they should not be misused by applying them to determining the best out of a candidate set of syntactic readings of some input. Rather, they can be expected to help at least with some types of the abovementioned failures.

Thus, example sentence (1), repeated below as (26), shows a syntactic irregularity in that the tense of *expect* should agree with that of *say* (the “sequence of tenses” rule in English) and be Past rather than Present. No syntactic repair will be necessary, however, because the semantic component of ontological semantics will not process *expects* as a tensed element but will assign its meaning to a timeless modality. In fact, one can probably speculate that this “deverbed” status of *expects* in the sentence allows for this syntactic laxness in the first place.

## 8.2 Building Basic Semantic Dependency

In Chapter 7, we illustrated, on the example of (1), repeated here as (26), the basic processes involved in generating TMRs.

- (26) Dresser Industries said it expects that major capital expenditure for expansion of U.S. manufacturing capacity will reduce imports from Japan.

The initial big step in semantic analysis is building basic semantic dependencies for the input text. Proceeding from the lexical, morphological and syntactic information available after the preprocessing stage for a textual input, on the one hand, and an empty TMR template, on the other, we establish the propositional structure of the future TMR, determine the elements that will become heads of the TMR propositions, fill out the property slots of the propositions by matching case role inventories and selectional restrictions on sets of candidate fillers. In this section, we will follow the flow chart in Figure 32 in describing the various processes involved in ontological semantic text analysis.

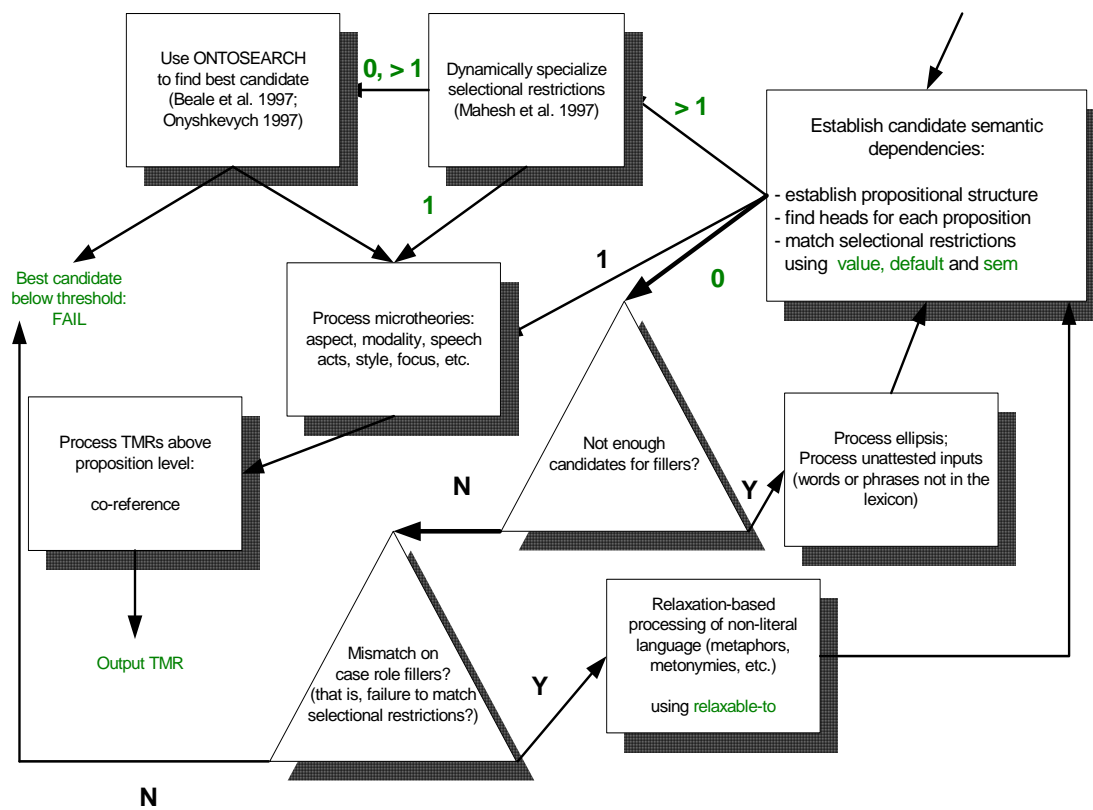


Figure 32. A Schematic view of the processes involved in the semantic analysis of text in ontological semantics.

### 8.2.1 Establishing Propositional Structure

(26) contains three syntactic clauses. A syntactic clause generally corresponds to a TMR proposition. A TMR proposition is the representation of a single predication, most commonly, an event instance. No TMR is well-formed if it does not contain at least one proposition. A TMR proposition is represented essentially as a template that combines a specification of the basic semantic dependency structure consisting of a head and its properties as well as such parameterized meaning components as aspect, time, modality, style and others. The boundary between parameterized and non-parameterized meaning will fluctuate with different implementations of ontological semantics. Different decisions concerning parameterization of meaning components will result in differences in the size and the content of the ontologies and lexicons involved (see Section 6.3 above).

It might appear paradoxical, then, that the TMR for (26) involves six, not three propositions (see 18). This means that no one-to-one correspondence exists between syntactic clauses and TMR propositions. There are six propositions in (26) because the SEM-STRUC zones of exactly six lexical entries contain ontological event instances. This is the simplest of the possible heuristics for establishing the propositional structure. In the case of (26), it was sufficient. As a result of the

decision to establish six propositions, six propositional templates are instantiated in the TMR working space, with the heads of the propositions filled by the six event instances and the rest of the property slots yet unfilled. It would have been easier to formulate the above heuristic in morphosyntactic rather than ontological terms: that a proposition should be instantiated in the TMR working space for each verb in the source text. Unfortunately, there is no isomorphism between syntactic verbs and semantic event instances. Indeed, there are four verbs in (26), *say*, *expect*, *manufacture* and *reduce*, but they engender only three propositions because, in our definition of TMR, the meaning of *expect* is parametrical. The other three propositions are engendered by nouns in (26), namely, those whose semantics is described by event instances: *expenditure*, *expansion* and *import*.

The choice of propositional structure can be complicated by the fact that some SEM-STRUCs can contain more than one event instance, e.g., the lexicon entry for the English *fetch* contains GO and BRING. This lexicon entry will engender two propositions in the TMR. In addition, ambiguity between word senses involving event instances and senses of the same word not involving them is a routine occurrence, especially in English, where virtually every noun can convert to a verb, e.g., *book*, *table*, *floor*, etc.

In practice, establishing propositional structure of natural language input proceeds by determining semantic dependencies among elements of input at different levels. Semantic dependencies are represented by having the dependent elements as fillers of slots in the governing elements. For example, OBJECT property values depend on OBJECT instances, OBJECT instances on EVENT instances in which they fill case roles, etc.

Once such basic dependencies are established, the remaining candidates for proposition headship are checked for whether their meanings are parametric, that is, whether they should be recorded not as proposition heads but rather as values of aspect, time, modality or other parameters inside the representation of propositions or TMRs. Once such parametric entities are accounted for, all remaining entities are declared heads of TMR propositions. We expect that such remaining material will include event instances and, more seldom, object instances. When such “free-standing” object instances are present, they become heads of propositions where the predication may be on a property value, as in (27), where this is established through syntactic clues) or, if there is no syntactic predication in the input, as in (28), with the implied meaning of existence. A special and interesting case is when the predication has the meaning of co-reference, which will be treated as illustrated in (29).

(27) The car is blue

(28) The blue car

(29) My son John is a teacher.

(27) and (28) get the same propositional structure, where the head of the proposition is the concept instance evoked by *car*, and BLUE is the (literal) filler of the property COLOR defined for CAR as a descendant of PHYSICAL-OBJECT. The difference in meaning between (27) and (28) is captured by the value of a parameter, the saliency modality, that is used in ontological semantics to deal with the phenomenon of focus, scoping over the filler of the slot COLOR in (30) and CAR in (31). (31) also illustrates how TMR treats existential quantification.

(30)

car-i			
	color	value	blue
modality-1			
	type	saliency	
scope		car-i.color	
	value	1	

(31)

car-i			
	color	value	blue
modality-1			
	type	saliency	
scope		car-i	
	value	1	

The index  $i$  means the  $i$ -th instance of the concept CAR in the TMR whose existence is posited. An object instance is assigned the head position in a TMR proposition as a stand-in in the absence of any event instances among the meanings of the clause constituents.<sup>75</sup> The corresponding rule for proposition head selection is, then: if one of the two open-class lexical entries in the input stands for an object and the other for its property, and there is no event involved, the object gets elevated to proposition headship.

The meaning (29) is represented as follows.

(32)

human-j			
	name	value	John
son-of-k			
	domain	value	*speaker*
	range	value	human-l
teacher-m			
	co-reference-n		human-j human-l teacher-m

Ontological description allows one to avoid introducing the separate concept TEACHER and instead refer to it always as the habitual agent of the event TEACH. Should such a decision be taken (and in the Mikrokosmos implementation of ontological semantics it was, in fact, not—see Section 7.1.5 above), the TMR would be as follows:

---

75. If the input were *John bought a blue car*, the lexical entry for *buy* (see Section 7.2 above) would instantiate the ontological event BUY, and that instance would assume the headship of the corresponding TMR proposition, obviating the need to elevate an object instance to proposition headship. The logic behind this decision is to avoid using dummy events as heads in TMR propositions. This desire is similar in motivation, though different in content, to Fillmore's (1968) proposal of elevating non-nominative cases to the Subject position in the absence of a more legitimate filler for the latter, thus making syntactic representations of sentences like *The door opened* similar to those for sentences like *John opened the door*.

(33)

human-j				
name	value	John		
son-of-k				
	domain	value	*speaker*	
	range	value	human-l	
teach-m				
aspect				
		iteration	multiple	
co-reference-n	human-j	human-l	teach-m.agent	

where the values of the properties of the aspect parameter carry the meaning of habituality.

There is no event instance involved in (29). This is similar to the cases (27) and (28). However, in (27) and (28) there is only one candidate for head. In the current case, there is no evidence in the input for selecting a single head from among the two OBJECT instances evoked by the words *teacher* and *John* or the relation instance evoked by the word *son*. Therefore, we posit that all three become heads of three (eventless) propositions, with the semantics of existential quantification. This outcome is predetermined by the fact that there is no way in which any one of the three elements can be “folded” into any other as a value of one of its properties—in contrast with the situation with events present, when instances of objects and relations are accounted for by filling the property slots of events, thus obviating the need to treat them as heads.

### 8.2.2 Matching Selectional Restrictions

The input to this stage of processing consists of the results of syntactic analysis of input and of the lexical look-up. For example, for the sentence (34), the results of syntactic analysis are in (35), while the results of the lexical look-up relevant to semantic dependency building are summarized in (36). In the specification of the lexical entries, direct use is made of the ontological concepts. (37) illustrates the relevant properties of the concept we need to use in explaining the process of matching selectional restrictions.

(34) John makes tools

(35)

root	manufacture		
cat	verb		
tense	present		
subject			
	root	john	
	cat	noun-proper	
object			
	root	tool	
	cat	noun	
	number	plural	

(36)

make-v1

```

syn-struct
  root      make
  cat       v
  subj      root  $var1
           cat   n
  object    root  $var2
           cat   n

sem-struct
  manufacturing-activity
           agent  value  ^$var1
           theme  value  ^$var2

```

```

John-n1
  syn-struct
  root      john
  cat       n-proper
  sem-struct
  human
           name  value  john
           gender  value  male

```

```

tool-n1
  syn-struct
  root      tool
  cat       n
  sem-struct
  tool

```

(37)

```

manufacturing-activity
...
agent  sem  human
theme  sem  artifact
...

```

The lexicon entry for *make* establishes that the meaning of the syntactic subject of *make* is the main candidate to fill the AGENT slot in MANUFACTURING-ACTIVITY, while the meaning of the syntactic object of *make* is the main candidate to fill the THEME slot. The lexicon entry for *make* refers to the ontological concept manufacturing-activity without modifying any of its constraints in the lexicon entry. This states that the meaning of its subject should be constrained to any concept in the ontological subtree with the root at the concept HUMAN; and the meaning of its object, to an element of the ontological subtree rooted at ARTIFACT. These constraints are selectional restrictions, and the lexicon entries for *John* and *tool* satisfy them.

Because the meanings of *John* and *tool* have been found to be dependent on the meaning of *make*, the semantic analyzer establishes that the instance of the event MANUFACTURING-ACTIVITY, listed in the lexicon as the semantic representation of the first sense of *make*, must be considered as the head of the proposition. As there is no other remaining material in the input, this is the only proposition.

Selectional restrictions in ontological semantics are used at all levels of building semantic dependencies—not just between predicates and their arguments but also between all the other pairs of governing and dependent elements in the input. In particular, adverbial meanings are folded into verbal (38) or adjectival (40) meanings, and meanings of nominal modifiers, including adjectives (39) and other nouns (41), are folded into that of the heads of noun phrases.

- (38) John makes tools quickly
- (39) John makes expensive tools
- (40) John makes very expensive tools
- (41) John makes power tools.

Unlike in the case of predicate-argument selectional restrictions, where, as we could see, the input offers both syntactic and semantic clues for matching, in the case of other modification, the system must rely only on semantics in deciding to which of the properties of the governing concept it must add a filler corresponding to the semantics of the modifier. Thus, in (39) above, it is only the meaning of the adjective that makes it a candidate filler for the COST property of the concept instrument while in *John makes large tools* the adjective will be connected on the property SIZE, while the syntax remains the same for both. As meanings of nouns typically do not correspond to properties, in cases like (41), even this clue is not available. This is the reason why the problem of nominal compounding is so confounding in English: *the IBM lecture* in (42) can mean a lecture given by IBM employees, a lecture sponsored by IBM, a lecture about IBM, a lecture given at IBM as well as many other things, which means that in different contexts the connection of *IBM* to *lecture* occurs on different properties (cf., e.g., Finin 1980, Isabelle 1984).

- (42) The IBM lecture will take place tomorrow at noon

In such cases there are several courses of action for the system to take, all costly. First, the system can look for prior co-occurrence of the meanings of IBM and lecture in the TMR, establish how they are connected and use this knowledge in resolving the current occurrence. If an antecedent is found, information in it may serve to disambiguate the current input. Thus, the information in the first sentence of (43) suggests that in the second sentence, *IBM* should be connected to *lecture* through the latter's LOCATION property. Of course, the heuristics for such disambiguation are error-prone, as, for instance, in the garden path case of (44).

- (43) John went to the IBM facility to give a lecture. The IBM lecture started at noon.
- (44) IBM sponsored a series of lectures on early computer manufacturers. Naturally, the IBM lecture was the most interesting.

### 8.2.3 Multivalued Static Selectional Restrictions

If the above lexical entry for *make* is used, (45) will violate selectional constraints, in that gorillas are not humans and, according to the lexical and ontological definition above (36, 37), are unsuitable as fillers for the AGENT slot of MANUFACTURING-ACTIVITY.<sup>76</sup>

- (45) The gorilla makes tools.

We know, however, that (45) is meaningful. To account for this, we must modify the knowledge sources. There are two ways in which this can be done. One could do this locally, in the lexicon

entry for *move* in (36), by changing the filler for the AGENT property to PRIMATE. It is preferable, however, to initiate this modification in the ontological concept MANUFACTURING-ACTIVITY. An immediate reason for that is that a meaning modification such as the one suggested above ignores the fact that most tools are manufactured by people. Indeed, this is the reason why most people would assume that *John* in (38) refers to a human male. It is in order to capture this knowledge that we introduced the ontological facet DEFAULT (see Section 7.1.1 above). The relevant parts of the ontological concept MANUFACTURING-ACTIVITY should become as illustrated in (46) while the lexical entry for *make* remains unchanged—no matter that it actually means a slightly different thing now.

(46)

manufacturing-activity

...			
agent		default	human
		sem	primate
...			

The semantic analyzer first attempts to match inputs against the fillers of the DEFAULT facet and, if this fails, against those of the SEM facet. If it succeeds, then the task of basic semantic dependency building is completed, and the system proceeds to establish the values of other components of TMRs (see Sections 6.2-5 above). Success in building basic semantic dependencies means that there remains, after the application of basic selectional restrictions, exactly one candidate word or phrase sense for every open-class input word or phrase. In other words, it means that the word sense disambiguation process has been successful.

Two more outcomes are possible in this situation: first, the basic procedure that applies selectional restrictions does not result in a single answer but rather returns more than one candidate word or phrase sense; second, none of the candidate senses of a word or phrase match selectional restrictions, and the basic procedure applying selectional restrictions returns no candidate senses for some words or phrases.

In both cases, the first remedy is to try to modify the selectional restrictions on the various senses so that a match occurs, and to do this in such a way as to minimize the overall amount of modification to the static knowledge. Such dynamic adaptation of selectional restrictions has not, to our knowledge, been proposed before. It is discussed in some detail below (Section 8.3.1).

An important methodological note is appropriate here. The many approaches to analysis using selectional restrictions imply the availability of ideal lexicons and other resources. Since discussions of selectional restrictions are usually centered around one example, such as *The man hit the colorful ball* in Katz and Fodor (1963), all that they require is to develop only a small fraction of the lexicon, and the constant temptation is to make the example work by presenting the senses exactly as needed for the example. If such discussions strove for any significant coverage of the

---

76. Of course, *John* can be understood as the name of anything, including a gorilla (cf. Schank 1975 about female fleas called John). However, there is a reasonable expectation that *John* is a person's name and, in the absence of evidence to the contrary, the system will be wise to stick to this expectation, while fully expecting that it is defeasible, in the spirit of abductive reasoning.

lexicon (see Sections 4.1 and 4.4 above), they would encounter serious practical difficulties having to do with the limitations and inaccuracies of resources, with complex trade-offs in the decisions taken while specifying different lexical entries and elements of their representations, and with maintaining consistency in the grain size of descriptions (see Section 9.3.6 below). As a result of these difficulties, the descriptions created for the purpose of illustrating a few selectional restrictions will very often fail when facing new selectional restrictions, for which they were not intended in the first place. In other words, the descriptions created for isolated examples are *ad hoc* and very likely to fail when significant coverage becomes a factor, which is always the case in practical applications. The goal of practical word sense disambiguation, then, is to eliminate as many inappropriate word senses in running text as possible, **given a particular set of static knowledge sources**.

The most common practical methods for resolving word sense ambiguities are based on statistical collocations (e.g., Gale *et al.* 1992, Yarowski 1992, 1995) or selectional restrictions between pairs of word senses. Of these two, the former is necessary when the method for word sense disambiguation does not rely on meaning representation (see Section 2.6.2.1 above) and extraction. Selectional restrictions provide stronger disambiguation power and, therefore, ontological semantics concentrates on selectional restrictions as the main disambiguation knowledge source, additionally so because we have acquired a source of selectional restriction knowledge of nontrivial size, viz., the ontology and lexicon complex.

However, neither a static ontology nor a static lexicon helps to achieve good disambiguation results all by itself. The real power of word sense selection lies in the ability to tighten or relax the semantic constraints on senses of a lexeme, or superentry, on the basis of choices made by a semantic analyzer for other words in the dynamic context. In other words, the selectional restrictions are not taken from the static knowledge sources directly but rather are calculated by the dynamic knowledge sources on the basis of both the existing static selectional restrictions and the interim results of semantic processing. Moreover, the resulting selectional restrictions are not recorded in the static knowledge sources, at least not until a method is developed for economically recording, prioritizing and indexing the entire fleeting textual and conceptual context for which they have been generated.

One often hears that context is crucial for semantic analysis. It is exactly in the above sense that one can operationalize this rather broad statement to make it practically applicable. Very few non-toy experiments have been carried out to investigate how this might be done in practice and on a realistic scale. Ontological semantics can be said to aspire to make realistic operational use of the notion of textual and conceptual context. We argue that:

- individual constraints between the head of a proposition and each of its arguments typically available in static knowledge sources (lexicons) are often not strong enough or too strong for effective selection of word senses;
- in addition to traditional selectional restrictions that check constraints between proposition heads and their semantic arguments, knowledge of constraints and conceptual relationships among the arguments of a proposition is critical because it is often not possible to determine a diagnostic context statically, i.e., before any decisions are made for the current sentence;
- effective sense disambiguation is helped by the availability of rich knowledge with a high

- degree of cross-dependence among knowledge elements;
- while representations such as semantic networks (including both simple labeled hierarchies, e.g., SENSUS (Knight and Luk, 1994) and ontological concept networks (e.g., the Mikrokosmos ontology (Mahesh, 1996; Mahesh and Nirenburg, 1995)) can capture such constraints and relationships, processing methods currently applied to semantic networks such as marker passing (e.g., Charniak, 1983; 1986) and spreading activation (e.g., Waltz and Pollack, 1985) do not facilitate selection of word senses based on dynamic context;
- marker passing and spreading activation are effective on well-designed and sparse networks but become less and less effective as the degree of connectivity increases (see Mahesh *et al.* 1997a,b for details).

### 8.3 When Basic Procedure Returns More Than a Single Answer

When the basic selectional restriction matching procedure returns more than a single candidate for each lexeme in the input, this means that the process of word sense disambiguation is not complete. The reason for that in this case is that the selectional restrictions are too loose. Additional processing is needed to bring the set of candidate word or phrase senses down to exactly one candidate for each lexeme, that is, to tighten the restrictions.

#### 8.3.1 Dynamic Tightening of Selectional Restrictions

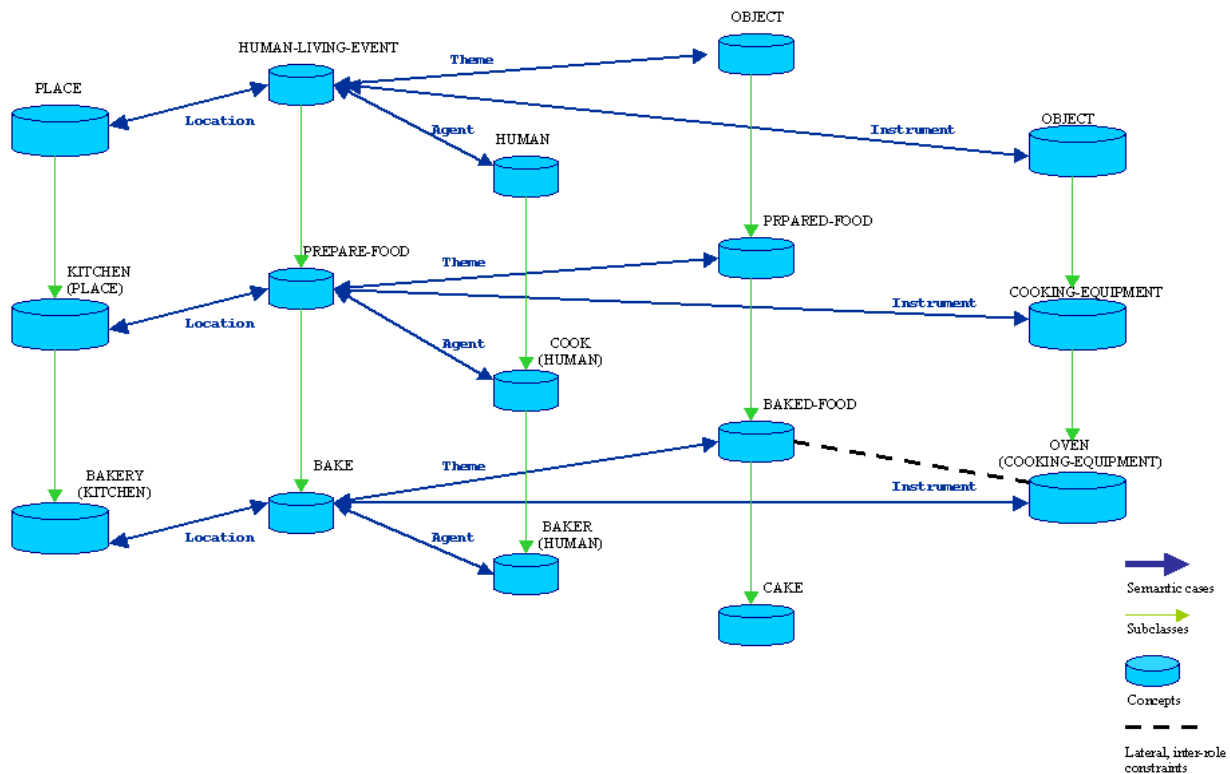
We will now demonstrate how dynamic tightening of selectional restrictions helps to resolve residual ambiguities. We will do this using the results of an experiment run in the framework of the Mikrokosmos implementation of ontological semantics, with the static and dynamic knowledge sources at a particular stage of their development (see Mahesh *et al.* 1997a for the original report).

Let us consider the sentence *John prepared a cake with the range*. Leaving aside, for the sake of simplicity, the PP-attachment ambiguity, let us concentrate on lexical disambiguation. In this sentence, several words are ambiguous, relative to the static knowledge sources. The lexical entry for *prepared* contains two senses, one related to the ontological concept PREPARE-FOOD, and the other, to PREPARE-DOCUMENT. The lexical entry for *range* has a number of different senses, referring to a mathematical range, a mountain range, a shooting range, a livestock grazing range as well as to a cooking device. In the latter sense, *range* can be related either to the ontological concept OVEN or the ontological concept STOVE. The lexical entry for *cake* is unambiguous: it has the ontological concept CAKE as the basis of its meaning specification. However, the ontological concept CAKE has two parents, BAKED-FOOD and DESSERT. The entry for *John* is found in the onomasticon and unambiguously recognized as a man's name. The entry for *with* establishes the type of relation on which the appropriate sense of *range* is connected to the appropriate sense of *prepare*. The possibilities include AGENT, with the filler that is a set (*Bill wrote the paper with Jim*), and INSTRUMENT (*Bill opened the door with a key*). The meanings of *a* and *the* will have "fired" in the process of syntactic analysis (see, however, Section 8.5.3 on additional meanings of the English articles related to the saliency modality).

After the sentence is analyzed according to the procedure illustrated in detail in Chapter 7 above, it will be determined that the meaning of *prepare* will become the head of the proposition describ-

ing the meaning of this sentence. The selectional restriction on *prepare* in the sense of PREPARE-FOOD matches the candidate constraint provided by the meaning of its direct object *cake* while the selectional restriction on *prepare* in the sense of PREPARE-DOCUMENT does not. This disambiguates *prepare* using only static selectional restrictions. *John*, in fact, matches either of the senses of *prepare*. So, while this word does not contribute to disambiguation of *prepare*, at least it does not hinder it.

Next, we establish that the correct sense of *with* is the one related to INSTRUMENT rather than AGENT because none of the senses of *range* are related to concepts that are descendants of HUMAN, which is a requirement for being AGENT of PREPARE-FOOD. At this point, we can exclude all those senses of *range* that are not compatible with the remaining sense of *with*, namely all but the two kitchen-related ones, whose meanings are related to STOVE and OVEN. Static selectional restrictions already disambiguated everything but the remaining two senses of *range*. No static selectional restrictions are available in the lexicon to help us complete the disambiguation process. We are now at the main point of our example, namely, a demonstration of the utility of dynamic selectional restrictions.



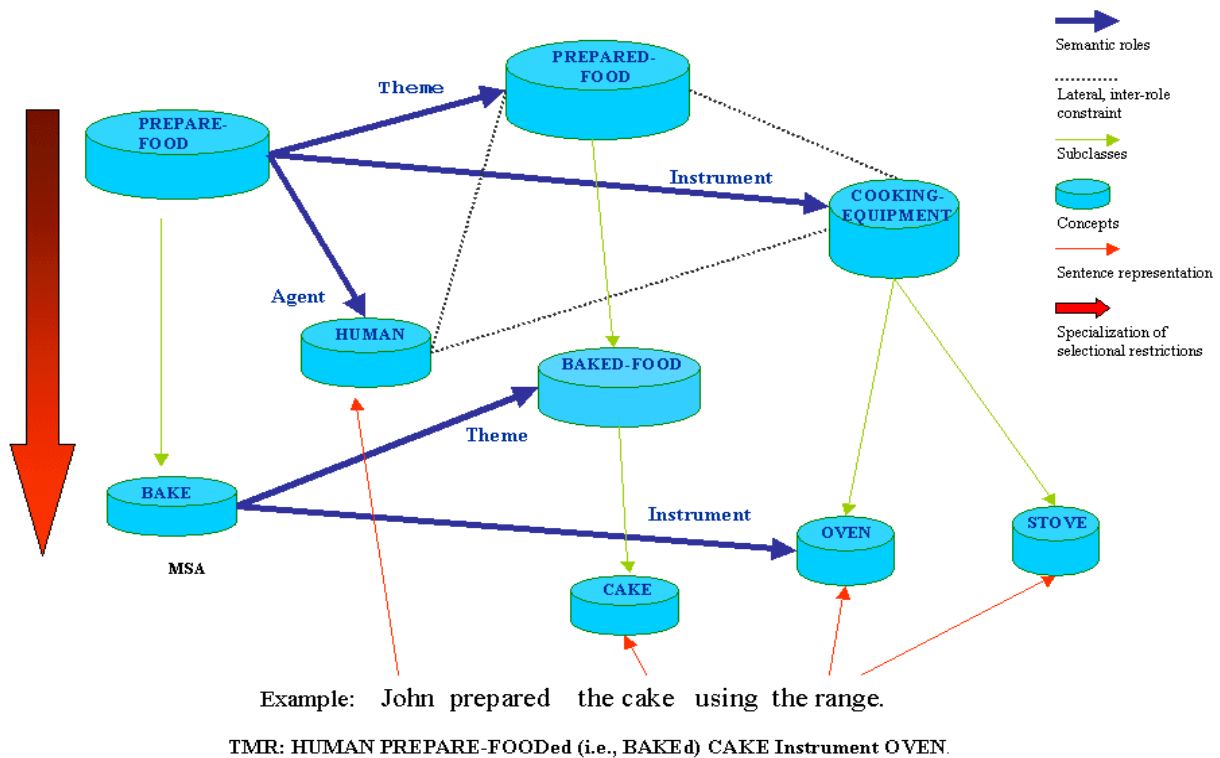
**Figure 33.** A fragment of the ontology showing main properties and constraints for PREPARE-FOOD. The properties are marked on blue arrows, their values marked on the color-coded circles representing concepts. The color coding underscores different inheritance chains. The values in parentheses refer to those in SEM facets, whereas the rest of the values are denote the fillers of DEFAULT facets in their respective properties.

As shown in Figure 33 (cf. Mahesh *et al.* 1997a), the ontological concept PREPARE-FOOD has PREPARED-FOOD as its THEME; COOK as its DEFAULT AGENT (and HUMAN as its SEM AGENT); and COOKING-EQUIPMENT as its INSTRUMENT. PREPARED-FOOD has many descendants, including BAKED-FOOD which, in turn, has many descendants, one of which is CAKE, the ontological concept defining the meaning of the English *cake* (or, for that matter, Russian *pirog* or Hebrew *uga*).

The last remaining task for disambiguation is to choose either OVEN or STOVE (signaled in the input by the corresponding word senses of *range*) as the THEME of the proposition head PREPARE-FOOD. Without context, this determination is not possible. However, once it is known that the THEME of this instance of PREPARE-FOOD is CAKE, a dynamic selectional restriction can be computed to make the choice. As CAKE IS-A BAKED-FOOD, it also meets the selectional restriction on the theme of *bake*. BAKED-FOOD is the THEME of BAKE, a direct descendant of PREPARE-FOOD, whose INSTRUMENT is constrained to OVEN but not STOVE. In order to make this disambiguation, we must, for the given context, specify PREPARE-FOOD as BAKE. In other words, we successfully dynamically apply the tighter selectional restriction on the INSTRUMENT of BAKE instead of whatever restriction is stated for the INSTRUMENT of PREPARE-FOOD. See Figure 34 for an illustration of this process.

An important point is that *bake* was not explicitly mentioned in the sentence. Nevertheless, once CAKE is determined to be a kind of BAKED-FOOD, the processor should be able to infer that the meaning of *prepared* should be, in this context, analyzed as BAKE since that is the only descendant of PREPARE-FOOD that takes BAKED-FOOD as THEME. This information is used by the procedure that computes dynamic selectional restrictions only after it is determined that the meaning of *cake* refers to BAKED-FOOD by virtue of CAKE being a descendant of BAKED-FOOD. Once this dynamic

context is inferred, the selectional restriction is tightened.



**Figure 34. Dynamic selectional restrictions in action. Specialization is needed, since checking selectional restrictions on PREPARE-FOOD retains the ambiguity between OVEN and STOVE, while the restrictions on BAKE lead to the desired disambiguation.**

The dynamic selectional restriction is necessary because one cannot realistically expect an English lexicon to contain a static selectional constraint associated with the INSTRUMENT role of PREPARE-FOOD that enables the system to distinguish between OVEN and STOVE, both direct ontological descendants of COOKING-EQUIPMENT, because any kind of cooking equipment can be an instrument of preparing food. Processing dynamic selectional restrictions is not a simple operation. Is it possible either to avoid it or at least to record its successful results in some way so that the next time a similar situation occurs, there would be no need for computing the restriction dynamically again?

One way of recording this information is to introduce yet another kind of selectional restriction—the inter-role **lateral** selectional restriction, which is not anchored at the head of a proposition but holds between two properties of the proposition head. Some lateral selectional restrictions, including the one between BAKED-FOOD and OVEN, are marked in Figures 33 and 34 with a dotted line. There is, of course, an alternative way that allows one to avoid introducing a new type of selectional restriction. The failure of dynamic selectional restrictions could trigger a request for adding to the ontology a direct descendant of a concept that will have the needed, tighter, selec-

tional restrictions. In other words, if BAKE were not already in the ontology and the English *range* required the disambiguation between OVEN and STOVE, this could trigger a request to add to the ontology a direct descendant of PREPARE-FOOD with the INSTRUMENT value of OVEN and the value of THEME, CAKE. In fact, there will be additional values in the various case roles of BAKE, but the above will “seed” the process of acquiring this concept.

It is reasonably clear that adding descendants to ontological concepts and recording lateral selectional restrictions in the ontology are different methods for doing essentially the same. At the same time, trying to avoid the processing of dynamic selectional restrictions by fixing the ontology statically involves the familiar time-space trade-off: if the information is not recorded, it will need to be computed every time a need arises. We also noted elsewhere (see Sections 5.3.1-2 above) that the occurrence in the input of a word with a specific type of ambiguity should not necessarily lead to further detailization of ontology.

Obviously, for given time-stamped ontology and lexicons, neither the appropriate descendants nor lateral selectional restrictions can be expected to be available for every input. In fact, NLP systems that depend on always having such information have not been successful in domain-independent word sense disambiguation because there is no way to establish the necessary grain size of description *a priori* and, therefore, any realistic NLP system must expect unattested elements in its input and have means for processing them (see also Section 8.4.3).

One must assume that knowledge sources for NLP are always incomplete and inaccurate, due to limitations of all acquisition methods as well as to unavoidable errors, including errors in judgment about grain size of description or a particular form that the description takes (see a discussion of synonymy in TMRs in Section 6.6 above). Our example showed how contextual processing, realized through dynamic selectional restrictions, helps to resolve the ambiguity even in the absence of complete background knowledge (such as a direct lateral selectional restriction between OVEN and BAKED-FOOD.)

Our example described a successful application of dynamic selectional restrictions. The reason for success was the presence of BAKE, that featured appropriately tight selectional restrictions, among the descendants of PREPARE-FOOD. Had BAKE not been available, the system would not have given up, though it would have taken a different route to the solution. This alternative solution would fail to resolve the ambiguity of *range* between OVEN and STOVE; it would accept this loss and fill the property of instrument for PREPARE-FOOD with the lowest common ancestor for OVEN and STOVE, namely, COOKING-EQUIPMENT. For many practical applications, this is an acceptable solution if, to put it plainly, the ambiguity is not important either for the text or the application. The former means that this information is accidental and not elaborated upon. This usually indicates that the corresponding concept instance is not likely either to be in the scope of a high-valued saliency modality filler in any proposition or to recur in many propositions in the TMR. An information item would not be important for a particular application if, for example, in MT, its translation is not ambiguous or there is no mismatch (e.g., Dorr 1994, Viegas 1997) between the source and target language on this word or phrase. In IE, importance can be judged by whether an information element is expected to be a part of the filler of an IE template slot.

Note that the main computational problem we are dealing with while trying to resolve the lexical

ambiguity of *range* is one of controlling the search for appropriate constraints, not the correctness of propagating those constraints that are already available from the static knowledge sources. Do we need to devise our own procedure for this purpose, or can such well-known computational methods as marker passing or spreading activation also accomplish this task? The answer depends on whether one can expect to solve this problem by using only heuristics based on the topology of the network, or also include the knowledge stored in the network. Marker passing and spreading activation, in their pure form, are too weak to guarantee that a selected context is the right one given all available knowledge. This is because these methods are adversely influenced by uninterpreted topological knowledge in the network that is not relevant to the current context. They do not reach into the semantics of the nodes and links.

As argued in detail in Mahesh et al. (1997a), in the case of marker passing, there may be paths of lengths equal or shorter than the one at which the procedure should aim, though not going through nodes in the desired context, such as BAKE. In Figure 33, for example, there is an alternative path from BAKED-FOOD to PREPARE-FOOD via PREPARED-FOOD, not via BAKE. This path consists of a THEME segment and an IS-A (SUBCLASS) segment just as the one going through BAKE. Thus, any choice in a marker passing algorithm will be hampered, as these two paths are equally preferable in this approach.

Let us follow the standard marker passing procedure on our example. The following nodes become the origins for marker passing: HUMAN, PREPARE-FOOD, the ontological concepts representing the other senses of *prepare*, the ontological concepts BAKED-FOOD, OVEN, STOVE and the ontological concepts representing the other senses of *range*. The goal of marker passing is to find the shortest path between each pair of origins. In pure marker passing, there are no weights on links; they carry a unit cost. Some candidates for shortest paths are illustrated in Figure 35.

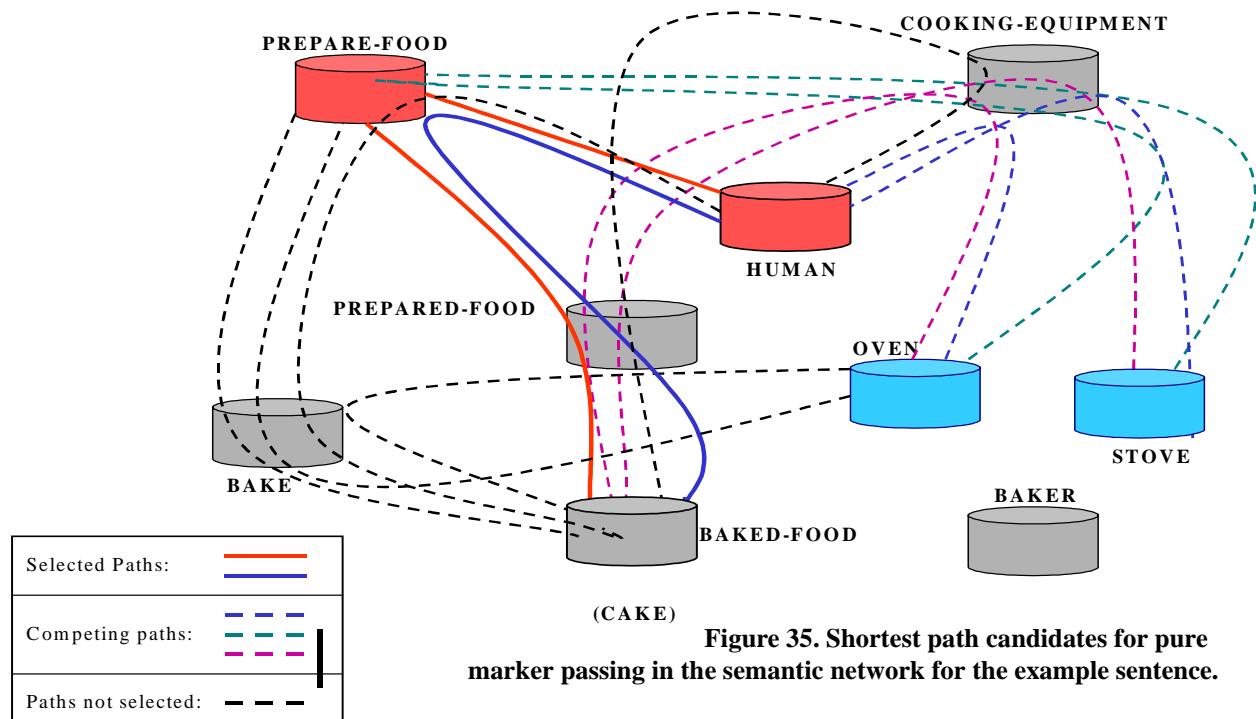


Figure 35. Shortest path candidates for pure marker passing in the semantic network for the example sentence.

It is clear from the figure that COOKING-EQUIPMENT and PREPARED-FOOD are strong intermediate nodes that could be chosen as elements of the path selected by the marker passing algorithm. BAKE might lose against these two and if so, the path from OVEN to BAKED-FOOD via BAKE may be rejected and the competing path via PREPARED-FOOD selected in order to maximize measures such as the total number of shared nodes among the selected paths. As a result, OVEN and STOVE turn out to be equally likely. Although BAKE had created a shorter path between OVEN and BAKED-FOOD than between STOVE and BAKED-FOOD, other parts of the network had an undue advantage over BAKE as a result of the above well-intentioned heuristics. In this situation, it is only by luck that OVEN might get selected, or even that the heuristics would discriminate between competing word senses sufficiently for any selection to take place at all.

We illustrated a small fragment of a conceptual network, with only a few types of available links listed. Any realistic model will have a much larger network with many other types of links between concepts, further decreasing the chances that the desired path through BAKE will be the least-cost path in the context of a sentence such as the one above. Moreover, these networks are almost always hand-coded and may include spurious links that eventually bypass certain desired paths. Processing mechanisms such as marker passing and spreading activation are simple and have a cognitive appeal, but their lack of reference to the content of the nodes makes them too weak for making the kinds of inferences needed for effective word sense disambiguation.

Our basic disambiguation method checks selectional constraints exhaustively, examining all the pairwise constraints on all word senses in a sentence, encoded statically in the ontological network or in the lexicon, using a very efficient search mechanism, called Hunter-Gatherer, based on constraint satisfaction, branch-and-bound, and solution synthesis methods (Beale *et al.* 1995, Beale 1997). To augment this method to process dynamic selectional restrictions, we introduce the Context Specialization Operator (CSO) with the following content: If a sense  $P$  is selected for a word  $w$ , and the rest of the word senses in the environment satisfy the constraints on  $P$ , examine the constraints on children of  $P$ ; if exactly one child  $C$  of  $P$  satisfies the constraints, then infer that the correct sense of  $w$  is  $C$ ; apply the constraints on  $C$  to other words.

The semantic analyzer checks selectional restrictions and applies the CSO iteratively, thereby resolving word sense ambiguities successively. Using the notion of CSO, the processing of our example sentence can be described as follows: CAKE is first determined to be a kind of BAKED-FOOD. Then, using this information, *prepared* is disambiguated to PREPARE-FOOD. Applying the CSO at this point shows that BAKE is the only ontological descendant of PREPARE-FOOD that satisfies the selectional restriction that the THEME must be BAKED-FOOD and the INSTRUMENT, one of the senses of *range*. Hence BAKE is included in the dynamic context, that is, the selectional restrictions have been dynamically tightened from those in PREPARE-FOOD to those in BAKE, and the latter's constraints are applied to *range*, thereby excluding STOVE and selecting OVEN.

The methods outlined above were implemented for semantic analysis in a Spanish-English MT system based on the Mikrokosmos implementation of ontological semantics. The system employed an ontology represented as a network of 5,000 concepts, where each node had an average connectivity of 16. A Spanish lexicon of about 37,000 word senses mapped them to nodes in this network.

It is certainly possible to fine-tune the ontological network or introduce and manipulate weights on the links to obtain a selection of OVEN over STOVE without resorting to dynamic selectional restrictions. However, such an approach does not guarantee that desired results will be obtained for inputs outside training corpora. Moreover, such tuning invariably has a catastrophic effect on processing other inputs. For example, if we fixed the network so that OVEN is somehow closer to BAKED-FOOD than STOVE, then OVEN would be selected even in an example such as *John ate the cake on the range*. There is, in fact, no information in this sentence that leads to a preference for either the STOVE or the OVEN sense of *range*. In general, these difficulties boil down to the following simple observation: any method that is oriented essentially toward manipulating uninterpreted strings does not have—and cannot be realistically expected to have—a sufficient amount of disambiguating heuristics for the task of text processing.

Statistical methods based on sense-tagged corpus analysis are subject to the same limitations as the network search methods. In a sufficiently general corpus, ample collocations of word senses may lead to irrelevant interference in sense disambiguation. For example, a high degree of collocation between the phrases *baked food* or *baked foods* or *bakery products*, on the one hand, and *oven*, on the other, helps to select the right sense of *range* in the example sentence. But just as with marker passing, the same statistical preference can mislead the processor into selecting the OVEN sense of *range* in *John ate the cake on the range*.

In general, any of the above disambiguating procedures, including those using dynamic selectional restrictions, may fail not because of their own faults but because the input is genuinely ambiguous.

### **8.3.2 When All Else Goes Wrong: Comparing Distances in Ontological Space**

When the procedure for applying dynamic selectional restrictions fails and the alternative solutions for some reason do not work either, for instance, because the lowest common ancestor of the candidate fillers for a property is judged too general, we can apply a technique that uses the ontology as a search space to find weighted distances between pairs of ontological concepts and thus to establish preferences for choice. Such a method, called Ontosearch, was developed in ontological semantics (Onyshkevych 1997) and applied in the Mikrokosmos implementation of ontological semantics (Mahesh *et al.* 1997a,b).

It is different from the standard marker passing and spreading activation techniques in that it uses the semantics of links and nodes in the ontological networks. Ontosearch is different from the procedure for applying selectional restrictions. The latter consists in simply determining that the candidate for filling a property slot in an ontological concept instance is a descendant of the ontological concept listed as a constraint there. Ontosearch undertakes to establish the weighted distance between the constraint and the candidate not only along the hierarchical (IS-A) backbone of the ontological network but following all and any links from every node—the node where the constraint originates (the constraint node), the candidate node and each of the intermediate nodes. Controlled constraint satisfaction in Ontosearch is managed by considering all relations and levying a cost for traversing any relations other than IS-A. The ontology is treated as a directed (possibly cyclic) graph, with concepts as nodes and relations as arcs. Constraint satisfaction consists in finding the cheapest path between the candidate concept node and the constraint nodes.

The cost assessed for traversing an arc may be dependent on the previous arcs traversed in a candidate path, because some arc types should not be repeatedly traversed, while other arcs should not be traversed if certain other arcs have already been seen. Ontosearch uses a state transition table to assess the appropriate cost for traversing an arc (based on the current path state) and to assign the next state for each candidate path being considered. The weight assignment transition table has about 40 states, and has individual treatment for 40 types of arcs; the other arcs (out of the nearly 300 total property types available in the ontology at the time when Ontosearch was first introduced) are treated by a default arc-cost determination mechanism.

The weights that are in the transition table are critical to the success of the method. An automatic training method has been used to train them (see Onyshkevych 1997). After building a training set of inputs (candidate fillers and constraints) and desired outputs (the “correct” paths over the ontology, i.e., the preferred relation), Ontosearch used a simulated annealing numerical optimization method (Kirkpatrick *et al.*, 1983; Metropolis *et al.*, 1953) for identifying the set of arc costs that resulted in the optimal set of solutions for the training data. A similar approach is used to optimize the arc costs so that the cheapest cost reflects the preferred word sense from a set of candidates.

Let us walk through a simple example of the operation of Ontosearch. Suppose, the ontological

semantic analyzer is processing the sentence: *El grupo Roche, a través de su compañía en*

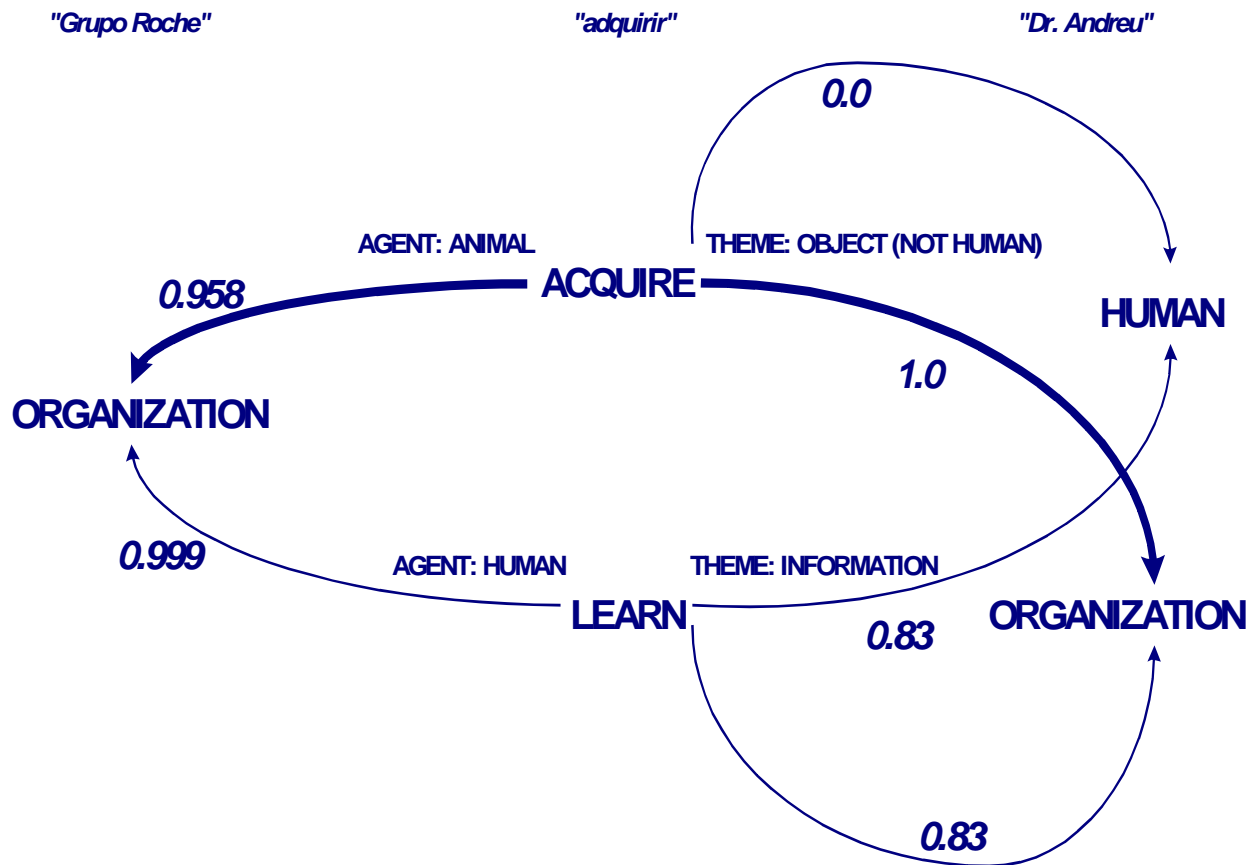


Figure 36. Applying

the general calculation of ontological distances for lexical disambiguation.

*España, adquirió el laboratorio farmacéutico Doctor Andreu, se informó hoy aquí,* ‘It was reported here today that the Roche Group, through its subsidiary in Spain, has acquired the pharmaceutical laboratory Dr. Andreu.’ We will concentrate on resolving just two potential ambiguities in this sentence. It is marginally possible to translate *adquirió* as *learned* in addition to the more common translation *acquired*. *Dr. Andreu* can be understood to refer to a company or to a person. Throughout the analysis, we assume that the static or dynamic selectional restrictions have not succeeded in disambiguating these cases.

A fragment of the ontological network used by Ontosearch to resolve the above ambiguities is illustrated in Figure 36. After the Ontosearch procedure has finished its operation, it has assigned the values for quality of transitions to the individual arcs (the higher the value, the more preferable the transition). In the figure, we can see that while ORGANIZATION (which is the conceptual basis of the meaning of *The Roche Group*) is a better candidate to fill the AGENT property of LEARN than of ACQUIRE, the fact that there is no penalty for having ORGANIZATION (the conceptual basis for one of the meanings of *Dr. Andreu*) fill the THEME of ACQUIRE while the somewhat awkward meaning of “learning an organization” represented by the path between LEARN and the ORGANIZATION sense of *Dr. Andreu* is penalized, so that the overall preferred path is the one highlighted in bold in the figure. Incidentally, “learning a person” gets the same penalty as “learning

an organization” while “acquiring a person” is simply prohibited in our ontological model of the world (see Section 7.1.2 above).

## 8.4 When Basic Procedure Returns No Answer

In the previous section, we considered the situation when the static selectional restrictions recorded in the lexicons and the ontology select more than one candidate from among the word senses of a word for each property of the proposition head. That introduces indeterminacy and a need for further disambiguation. In this section, we are considering the opposite situation—when a selectional restriction fails to find any candidate for filling the value of a property. There can be two reasons for such a contingency: the candidate lexeme is available in the lexicon but has no sense that matches the selectional restriction, or there is no recognizable candidate in the input on which a match attempt could be made. The former case involves either what is known in the philosophy literature as sortal incongruity, or incorrectness (e.g., Thomason 1972) or the use of non-literal language. There are also two possible reasons for a candidate being unavailable: ellipsis or the presence of unattested words or phrases in the input.

### 8.4.1 Relaxation of Selectional Restrictions

We are already familiar with the use of the facets DEFAULT and SEM (see Sections 6.2 and 7.1.1 above). Thus, for instance, PREPARE-FOOD has COOK as the value of its AGENT property on the DEFAULT facet and HUMAN on the SEM facet. Unlike in example (45), the use of GORILLA as a candidate for the filler of AGENT of PREPARE-FOOD cannot be accommodated by the constraint in the SEM facet: all primates make tools but not all primates cook. Nevertheless, the sentence *The gorilla cooked dinner* can be given an interpretation by using the facet RELAXABLE-TO on the AGENT property of PREPARE-FOOD.

This facet is the main resource for dealing with the case when no sense of an available lexeme matches a selectional restriction. The sentence *The baby ate a piece of paper* illustrates a typical case of sortal incongruity: in ontological semantics, this is reflected in the fact that INGEST, the ontological basis of the meaning of *eat*, requires a descendant of EDIBLE as a filler of its THEME. PAPER is not a descendant of EDIBLE, it is a descendant of MATERIAL. The facet RELAXABLE-TO ensures that this meaningful sentence obtain its interpretation.

### 8.4.2 Processing Non-literal Language

A similar relaxation technique is used to accommodate non-literal language. Non-literal language is understood in ontological semantics as having lexemes carry derivable but unrecorded senses. For example, in the sentence *The pianist played Bach*, the selectional restriction on the SEM facet of the THEME property of PLAY-MUSICAL-INSTRUMENT, the concept on which the appropriate meaning of *play* is based, is MUSIC-COMPOSITION, which is the basis for specifying the meaning of such English words as *sonata*, *concerto*, *symphony*, etc. The entry for *Bach* in the onomasticon characterizes it as HUMAN. The discrepancy in the selectional restriction is due to fact that the filler of the theme property is realized as a standard metonymy of the ‘author for creation’ type. In the case of metonymy, the same simple treatment that we used for the case of sortal incongruity will not work.

The difference between treating sortal incongruity and metonymy is that in the former case the analyzer, after establishing a match between the candidate filler concept and the selectional restriction on the RELAXABLE-TO facet for a property, directly fills the corresponding slot of the TMR concept with an instance of this same candidate filler concept. In the case of metonymy, the match takes place similarly to the above case, but what becomes the filler of the property in TMR is the instance of a different concept. This concept, the expansion of the metonymy, cannot be derived dynamically in the current microtheory of non-literal language processing used in ontological semantics. Until and unless such a theory becomes available (and it is not at all clear whether such a theory is, in fact, feasible—see Fass 1991, Barnden *et al.* 1994, Onyshkevych and Nirenburg 1994, Beale *et al.* 1997), a stop-gap measure is to directly list the expansions of metonymies in the static selectional restrictions, namely, in the RELAXABLE-TO slots of corresponding properties.

The facet RELAXABLE-TO, when used for treating non-literal language, will necessitate a modification to the format of ontological specification beyond the level in extant implementations of ontological semantics. When applied to the THEME of PLAY-MUSICAL-INSTRUMENT, in order to account for metonymies such as that in *The pianist played Bach*, the relaxable-to facet will have to refer to both the literal interpretation that will be needed for matching the input and the expansion that is needed to include the appropriate meaning in the TMR:

```

play-musical-instrument
...
  theme          sem          musical-composition
                  relaxable-to match          human-1
                              expansion        musical-composition
                                                    composed-by value human-1.name
...

```

The analyzer will fail to match the SEM selectional restriction and will proceed to the RELAXABLE-TO one. Here it will make a match on the value HUMAN and proceed to instantiate the concept MUSICAL-COMPOSITION with its COMPOSED-BY property filled by the same named instance of HUMAN (marked as coindexical in the ontological specification of PLAY-MUSICAL-INSTRUMENT). The property COMPOSED-BY has as its domain LITERARY-COMPOSITION, in addition to MUSICAL-COMPOSITION.

The ontological semantic analyzer will carry out more work on the sentence *The pianist played Bach* than described above. This is because the English *play* has another sense, the one related to sports. It is represented using the ontological concept SPORTS-ACTIVITY, the AGENT property of which (the meanings of both the subject and the direct object of *play* will be connected on the AGENT property of the concept SPORTS-ACTIVITY) has the selectional restriction that matches HUMAN, among other concepts, e.g., TEAM. The analyzer will prefer the musical reading of the sentence because the default value of AGENT of PLAY-MUSICAL-INSTRUMENT will be matched by the meaning of *pianist*, namely, MUSICIAN, while the latter will not be a DEFAULT value of AGENT for the SPORTS-ACTIVITY sense of *play* (it will match the SEM facet). This underscores, again, the general rule that DEFAULT constraints have priority over SEM constraints which, in turn, are preferred to the RELAXABLE-TO constraints. Let us not forget that, as always, this analysis may be overturned by text-level context.

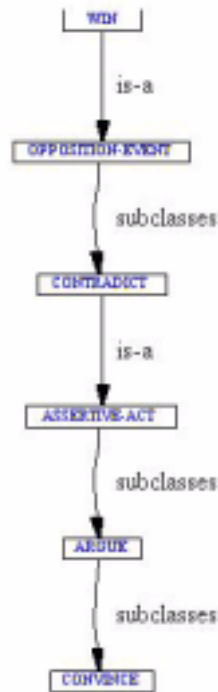
If the example sentence is followed in a text, as in the well-known joke, by *Bach lost*, the analyzer will have dynamically to revise the preferences derived during the processing of the first sentence due to the requirements of text coherence (captured in ontological semantics, still only partially in the extant implementations, through discourse relations in TMRs—see Section 8.6.3 below). The second sentence makes sense only if the overall context is sports. The analyzer (possibly, in a simplification) follows the rule that a text belongs to a single conceptual context or domain (cf. Gale and Church about one sense per discourse). This rule is triggered in this example because the second sentence is elliptical (see Section 8.4.4 below for the ontological semantics take on ellipsis processing), and for elliptical sentences there is a strong expectation that they describe another component of the same complex event whose description was begun in earlier sentence(s). The clue is especially strong if this sequence of sentences is contiguous. One of the factors contributing to our perception of the text as humorous is that people who analyze it follow the same path of “priming,” that is, selecting a particular complex event and expecting to stay with it in the immediate continuation of a text (for additional factors dealing with juxtaposing the primed event on the competing one, see Raskin 1985, Attardo and Raskin 1991, and Attardo 1994).

The switch to the different sense of the event in the above example occurred in a situation where that sense was already recorded in the lexicon. When input contains metaphoric language, the other kind of non-literal language processed by ontological semantics, such a switch must be made without the benefit of a previously recorded sense. Consider the sentence *Mary won an argument with John*. No sense of *argument* matches the selectional restrictions on the THEME of WIN which are MILITARY-ACTIVITY, SPORTS-ACTIVITY and GAME-ACTIVITY. If the selectional restriction on the RELAXABLE-TO facet of the THEME property of WIN matches ARGUE, this case can be treated as metonymy. It is more interesting, however, at this point, to consider a situation in which the selectional restriction on the RELAXABLE-TO facet of the THEME of WIN does not have a value. It is in this situation that the analyzer must process a metaphor, which in our environment means searching for an event whose selectional restrictions match the fillers of the case roles in the proposition obtained from the above sentence. Specifically, for this example, such an event should match the selectional restriction HUMAN on the fillers of the AGENT (*Mary* and *John*) properties and ARGUE on the THEME property of WIN. One good solution would be the concept CONVINCE: the sentence *Mary convinced John in an argument* is indeed a non-metaphorical rendering of the original example. Unfortunately, there is no theory of metaphor, in ontological semantics or elsewhere, that is capable of guaranteeing that such a result could be procedurally obtained.

A microtheory of metaphor in ontological semantics would need to search through the entire set of events looking for matches on inverse selectional restrictions. This must be done in an efficient manner. If the algorithm is designed to check this search space exhaustively (discarding only those candidates that at any given moment can be proved not to fit the bill), then it is likely that it will return more than one candidate solution. Then a special routine will have to be written to establish a preference structure over this set of candidate solutions, which is not a trivial task. If, however, the algorithm is designed on the basis of satisficing, that is, if it will halt when the first appropriate candidate is found, the main issue, which may be equally complex, becomes how to establish the satisficing threshold so as to diminish the probability of an erroneous choice.

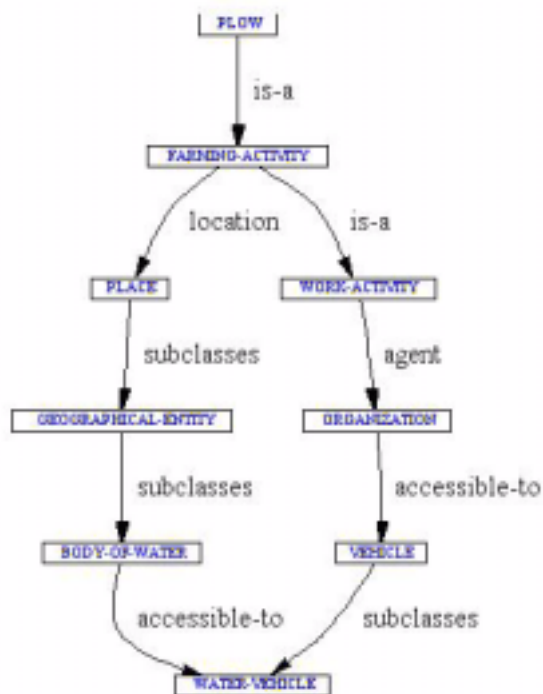
Intuition suggests that the best strategy for fitting the inverse selectional restrictions to the events is by relaxing the restrictions in the events themselves, that is, by moving from the source domain

of the metaphor, the origin of the search, toward the root of the ontological tree. Even a cursory manual examination of several examples immediately shows that such hopes are unjustified. Indeed, continuing with the assumption that CONVINCe is a good literal substitute for WIN in the above example, we can see in Figure 37 that the most economical path between the two concepts in the ontology is multidirectional.



**Figure 37. The most economical path between the two concepts in the ontology may be multidirectional**

In *The ship plowed the waves*, the path between the metaphorical PLOW and the literal MOVE-WATER-VEHICLE is even more convoluted (see Figure 38).



**Figure 38. The convoluted path between the metaphorical PLOW and the literal MOVE-WATER-VEHICLE.**

Whether the hope for the microtheory of metaphor in ontological semantics lies in figuring out how to navigate such paths or in applying other algorithms, the best ontological semantics can do at this point is to define the problem and the search space in which to look for an answer. It is clear then that until such a microtheory is available, it is advisable to reduce metaphor to metonymy by specifying fillers for RELAXABLE-TO facet values of event properties in the ontology, whenever possible.

### 8.4.3 Processing Unattested Inputs

Used in real-life conditions, any NLP system must expect inputs that contain words and phrases for which there are no entries in the lexicons or the onomasticons. Such inputs fall into several categories. In certain text types, most prominently, in journalistic prose, one should expect proper names to form the largest single category of unattested input elements. The preprocessing component of the analyzer (see Section 8.1 above) contains routines for recognizing unattested proper names. As they are used at an early stage in the processing, such routines use only textual context elements as clues—for instance, if a phrase ends in *Inc.*, *GmbH*, *Corp.*, *Cie*, *NA* or *Ltd.* it is the name of a company, and so on.

The unattested material that is not recognized and categorized as a kind of proper name is also processed by the special routine that uses the available morphological, syntactic and semantic analyzers to assign as many features to the unknown word as is possible when no lexicon entry is there. Morphologically, this routine attempts to assign a part of speech and other grammatical fea-

tures (such as gender or person) to the unknown word on the basis of its form as well as its syntactic context. Syntactically, it establishes this word's or phrase's position in the syntactic dependency structure generated by the syntactic analyzer for the input text. Semantically, the procedure uses the syntactic dependency and the knowledge available in the lexicon entry to link syntactic and semantic dependencies to weave the meaning of this word into the TMR. Humans perform exactly the same operations, quite successfully, when faced with texts like Lewis Carroll's *Jabberwocky*: "Twas brillig and the..."

As the meaning of an unattested word is not reliably available, the procedure does its best to constrain this meaning by assuming that, when the word is a semantic modifier, the selectional restrictions on the properties that the unknown word must match to be the appropriate filler in the semantic dependency structure define the meaning of the unattested word. When the unknown word is a semantic head, the selectional restrictions in its lexicon definition will exactly match the constraints on the senses of elements that fill the corresponding properties in the TMR for the sentence in which the unattested word appears. As we demonstrated in Section 8.2.2 above, the algorithms for processing selectional restrictions involve matching two values—that of the constraint on the property and that of the candidate filler for that property. In the 'regular' case, this is, then, reciprocal matching. When unattested words occur, one of the values for the match is unavailable, so that the match is trivial and always succeeds, as it is a match of a constraint against a general set of possible candidates. Let us consider, first, an example of processing an unattested modifier and then, that of an unattested head.

Thus, in the sentence *Fred locked the door with the **kheegh***, the highlighted string is an unattested word. Its position between a determiner and the end of the sentence easily identifies it as a noun. The prevalent sense of *with* combined with the availability of the INSTRUMENT property in the meaning of *lock*, links the meaning of *kheegh* to the filler of this property in the concept LOCK-EVENT. The selectional restriction on INSTRUMENT in LOCK-EVENT is KEY on the DEFAULT facet and ARTIFACT on the SEM facet. At this point, a TMR emerges, whose relevant part is shown in (47). The filler of the INSTRUMENT property is an instance of ARTIFACT, which means that the procedure used the SEM constraint of the property rather than committing itself to the DEFAULT constraint (and using the concept key in the TMR) on insufficient evidence—after all *kheegh* may mean 'credit card.' A side effect of this processing is that a tentative lexicon entry for *kheegh* (48) can be automatically constructed with the content determined by the above results.

(47)

...

```
lock-event-6
  agent      value human-549
  theme      value door-23
  instrument value artifact-71
  ...
```

(48)

```
kheegh-n1
  syn-struct
    root    kheegh
    cat     n
  sem-struct
    artifact
      instrument-of  value lock-event
```

Now consider the sentence *Fred **launched** the door with the key*. A lexicon entry will be created for the unattested event with selectional restrictions provided by the meanings of the case role fillers:

```
lauch-v1
  syn-struct
    root    lauch
    cat     v
    subject root    $var1
           cat     n
    object  root    $var2
           cat     n
    oblique root    with
           cat     prep
           object  root    $var3
           cat     n
  sem-struct
    event
      agent    value    ^$var1
              sem     human
      theme    value    ^$var2
              sem     door
      instrument value    ^$var3
              sem     key
```

The above means that the event realized by *lauch* has a human agent, a theme that is a door and an instrument which is a key. This is all the information that can be reliably gleaned from the input sentence. While it is not expected that the lexicon entry can be completed without an inspection and further tightening by a human knowledge acquirer, recording the results of processing unattested input reduces the amount of manual acquisition work.

#### 8.4.4 Processing Ellipsis

Sometimes the basic procedure for processing selectional restrictions returns no result because the input does not contain a sufficient supply of candidates for filling the case roles of the proposition head, e.g.:

- (49) Nick went to the movies and Paul to the game
- (50) I finished the book
- (51) The book reads well
- (52) John shaved

(49) is probably the most standard case of syntactic ellipsis, where the second clause follows the syntactic structure of the first and does not repeat a certain word, in this case, the verb. Most of the literature on ellipsis in theoretical and computational linguistics concentrates on this, symmetrical, type of ellipsis. But it is clearly not true that ellipsis is an exclusively syntactic phenomenon (Baltes 1995 and references there). Examples (50) - (52) are not elliptical syntactically and, in fact, many natural language processing programs (or theoretical linguists) will not treat them as elliptical. From the point of view of ontological semantics, however, some of them are. In each of the three examples, the failure to match a selectional restriction due to the lack of lexical material in the input to fill a case role, signals the need for processing semantic ellipsis. Analysis of (50) must involve instantiation of an ontological concept not directly referred to in the input, namely, *read* or *write* (or, at a stretch, *bind* or *copy*). Similarly, there is no lexical element in (51) that can be considered as a candidate filler for the AGENT property of the meaning of *read*. *Shave* in (52) is the intransitive sense of the verb. In ontological semantics, however, the transitive and intransitive senses of the verb *shave* are defined in terms of one concept. This concept expects an AGENT and a PATIENT. In the surface form of (52) there is no separate candidate for the filler of PATIENT, after the meaning of *John* is selected to fill the AGENT slot. However, in the lexicon entries for all reflexive verbs, we record that the meaning of the single NP constituent that they require fills both the property of AGENT and of PATIENT. The intransitive sense of *shave* is treated as a reflexive verb, making semantic ellipsis in this example illusory.

Semantic ellipsis is often triggered by the occurrence of a verb like *finish* in (50). This verb belongs to a class of verbs that take other verbs as their complements. In their lexicon entries, these verbs require an EVENT as the filler of their THEME property. Moreover, in some cases such verbs constrain the semantics of their themes, which, obviously, helps to recover their meanings when in the input text the verbs corresponding to these events are elided, as in the example. When it is not possible to impose a strong constraint on the filler of THEME, the recovery procedure is more complex. For example, the THEMES of *enjoy* in sentences *Mary enjoyed the movie*, *Mary enjoyed the book* and *Mary enjoyed the cake* can be recovered as SEE, READ and INGEST. This is because the ontological concepts for *movie*, *book* and *cake* contain the above concepts in the DEFAULT facet of their THEME-OF property. Similarly, the example we briefly referred to in Section 3.4.2 above, *fast motorway*, is treated as a regular case of ellipsis: the missing event DRIVE is recovered as the filler of the DEFAULT facet of the property LOCATION-OF on the concept ROAD which is the basis of the meaning of *motorway*. The meaning of *fast* is a region on the scale that is the range of the property VELOCITY on the concept DRIVE.

The ontological concept for *lizard*, however, does not contain a DEFAULT value in its THEME-OF property because there is no typical EVENT that can be enjoyed concerning lizards. This makes the recovery of the ellipsis in *Mary enjoyed the lizard* a more difficult task: is the required event INGEST or SEE or something else? The natural procedure here is to weaken the constraint on the EVENT by defining it as belonging to the ontological subtree rooted at the filler of the SEM facet of the THEME-OF property of LIZARD. If the EVENT contains a set of values, the procedure will use the lowest common ancestor of all of them in the ontology. This makes it clear that the treatment of this kind of semantic ellipsis has a great deal in common with the treatment of unattested verbs. In both cases, the semantics of the EVENT realized by the verb, either elided or unattested, is determined, to the degree possible, by the constraints on the content of the inverse case role properties (THEME-OF, INSTRUMENT-OF, AGENT-OF, etc.) in the meanings of the arguments of these verbs.

Some of the verbs that trigger semantic ellipsis have additional senses that are not elliptic. Thus, *I finished the bench* is genuinely ambiguous between the non-elliptic sense ‘I covered the table with varnish’ and the elliptic sense ‘I finished making/repairing/painting/... the bench.’ Such cases must be treated both as potentially ambiguous and potentially elliptic. This means that the procedure that matches selectional restrictions must expect at the same time to obtain the state of affairs with more than one candidate solution (if the input is to be treated as ambiguous) or no candidate (the case of ellipsis). As the above eventualities are quite frequent, the procedure becomes quite complex.

## 8.5 Processing Meaning Beyond Basic Semantic Dependencies

When selectional restrictions are matched successfully and, thus, the basic semantic dependency for an element of input is established, it is time to establish the values of the various parameters defined in TMR, both alongside basic semantic dependencies within a proposition and alongside propositions in the TMR for an entire text (see Example (18) and Section 6.3 above). Each propositional parameter characterizes a specific proposition; it has a set of values that contribute standardized meaning components and belong to instances, but not to ontological concepts.

Suprapositional parameters characterize an entire TMR. They come in three varieties. The first type involves instantiation of ontological relations with propositions filling their DOMAIN and RANGE slots. In other words, it establishes relations among propositions. The second type groups TMR elements from different propositions according to the semantics of the particular parameter, for example, into CO-REFERENCE chains or into a partial ordering of time references. The third type of suprapositional parameter is given a value through the application of a function over the values of specific propositional parameters; this is the way in which the STYLE of an entire text is calculated on the basis of style values generated for individual propositions.

In what follows, we describe the specific parametric microtheories that have been developed for the Mikrokosmos implementation of ontological semantics. There may be other implementations, based on different approaches to building the specific microtheories (see Section 1.7 above for a discussion of the microtheory approach). In other words, the microtheories may, in principle, be replaced with other, better, microtheories at any time and, we believe, with a minimum of disturbance for the entire complex of static and dynamic knowledge sources in an ontological-semantic application. The emphasis in this section is on the content of the semantic microtheories and nature of clues for assigning values of properties defining the microtheory and not on the many ways in which languages express the meanings captured by the various parametric microtheories.

### 8.5.1 Aspect

Aspectual meanings in ontological semantics are represented using a set of two properties—PHASE and ITERATION. PHASE has four values—BEGIN, CONTINUE, END and BEGIN/CONTINUE/END. The latter value covers events which are perceived as momentary on a human-oriented time scale. Technically, of course, these events will, in fact, have duration, albeit a very short one (see Comrie 1976: 41-44 for an attempt to analyze this distinction at a finer grain size; we believe that this would serve no useful purpose in ontological semantics). ITERATION, which, predictably, refers to repetitiveness of a process, is represented using an actual number or the indefinite value MULTIPLE. The meaning of PHASE refers to the temporal stage in a process—whether the input

talks about the initial (BEGIN) or the final (END) stage or about neither (CONTINUE).

Table 14: Clues for Assignment of Aspectual Values

Phase	Iteration	Examples
begin	1	Ivan zapel ‘Ivan started singing’
begin	multiple	Obychno Ivan nachinal pet’ ‘Usually, Ivan started singing’
end	1	Ivan dostroil dom ‘Ivan finished building the house’
end	multiple	Ivan stroil po domu kazhdyj mesjac ‘Ivan built a house every month’
continue	1	Ivan sidel na skam’e ‘Ivan sat on a bench’ or ‘Ivan was sitting on a bench’
continue	multiple	Ivan sidel na skam’e po sredam ‘On Wednesdays Ivan sat on the bench’
b/c/e	1	Ivan vyigral gonku ‘Ivan won the race’ Ivan vyigryval gonku odin raz ‘Ivan won the race once’
b/c/e	4	Ivan vyigral gonku chetyre raza Ivan vyigryval gonku chetyre raza

The examples in Table 14 show that clues for assignment of aspectual values in our microtheory will, in the general case, be composite. This finding corroborates the conclusions one can reach from the material presented in Comrie (1976) that a given morphological marker of aspect in a language does not necessarily predict the aspectual meaning of a proposition. For example, in the last two rows of the table above, Russian verbs with different morphological aspectual markers contribute to the same semantic value of ASPECT (that is, to the same combination of values of PHASE and ITERATION).

The microtheory of aspect proposed here is not the first one used in ontological semantics (e.g., Pustejovsky and Nirenburg 1988). In earlier implementations, aspect was described using a super-set of the properties we use here. In particular, the properties of duration and telicity were used in addition to PHASE and ITERATION. Duration distinguished momentary and prolonged events (for example, *he woke up* vs. *he slept*). Telicity distinguished between resultative and non-resultative events (for example, *he built a house* vs. *he slept for ten hours*).

As the main motivation for parameterizing a component of meaning is economy of ontological knowledge acquisition (see Section 6.3 above), it is only worth our while to parameterize duration or telicity if there exist a sufficient number of pairs of lexical items (possibly, different senses of the same word) whose meanings differ only in the values of these parameters. In such a case the meaning of N such pairs (2N lexical items) could be expressed with, at most, N ontological concepts plus the values of one TMR parameter. The alternative, non-parameterized approach may lead to up to 2N ontological concepts. In the case of duration, we have failed to detect any significant body of event realizations that feature such a dichotomy. Whatever examples of variation of

duration there actually exist (e.g., the momentary *he sat down* vs. the prolonged *he sat for an hour*) can be readily captured by the appropriate values of the PHASE parameter—BEGIN/CONTINUE/END and CONTINUE, respectively.

Telicity, similarly, does not seem to warrant parameterization. While the phenomenon of telicity is real enough, and information about resultativity of an activity should be included in the EFFECT property of ontological events (see Section 6.7 above),<sup>77</sup> once again, we do not see a critical enough mass of pairs in the lexical stock of many languages to suggest parameterization of this meaning component.<sup>78</sup>

In what follows, we illustrate the assignment of aspectual values in the microtheory of aspect in the Mikrokosmos implementation of ontological semantics for analyzing English. Of course, in analyzers for other languages there may be additional kinds of clues (e.g., verbal prefixation in Slavic languages, as in the Russian *zapel*). Still, English examples are sufficiently representative. First, there is a class of what we would call phasal verbs—*begin, cease, commence, stop, finish, desist from, carry on, keep, continue*, etc.—whose contribution to the overall meaning of the sentence in which they appear is aspectual. The aspectual value for the proposition in which a phasal verb like *begin* appears will be obtained from the SEM-STRUC zone of the lexical entry for the appropriate sense of *begin*:

begin-v2

syn-struct			
root	begin		
cat	v		
subj	root	\$var1	
	cat	n	
xcomp	root	\$var2	
	cat	v	
	obj	root	\$var3
		opt	+
sem-struct			
event			
	agent	value	^\$var1
	theme	value	^\$var3
aspect			
	phase	begin	

The ASPECT property in the SEM-STRUC of *begin-v2* appears at the level of proposition whose head (marked as ^\$var2) is the meaning of the (syntactic) head of the infinitival or gerundive construc-

---

77. The property of effect in the ontological description of events helps to cover a wide variety of important phenomena, such as causality, entailment and many others, including telicity. Thus, ontological semantics does not require any special device for representing telicity in the lexicon, as proposed by Pustejovsky 1995:99-101.

78. Many studies of aspect (e.g., Comrie 1976:44-48; Vendler 1967:102-104, where telicity was referred to as ‘accomplishment’; Klein 1974; Dowty 1972; Verkuyl 1972, 1993) have difficulties establishing telicity as a feature distinct from completion. For us, this means that this feature does not have a *bona fide* semantic aspectual significance (see Section 8.5.3 below for further discussion).

tion occupying the xcomp position in the syntactic dependency of *begin-v2*, that is, the meaning of *sing* in *John began to sing*. Phasal verbs do not have any meaning other than aspectual. The next example illustrates how phasal value can be contributed not by a special verb but by a closed-class lexical morpheme (either free, a preposition or a particle, or bound, an affix). In this case, the word governing the closed-class morpheme contributes a non-aspectual meaning to the TMR. The example below is the English phrasal verb *drink up* that combines the non-aspectual meaning INGEST with the phasal value END and iteration meaning 1. The lexicon entry treats *drink up* as one of the senses of *drink*, specifically, the one subcategorizing for the literal *up* rather than for the category of preposition. The direct object of this verb is optional—both *Drink up!* and *Drink your milk up!* are well-formed.

```

drink-v23
  syn-struct
    root    drink
    cat     v
    subj    root    $var1
           cat     n
    obj     root    $var2
           cat     n
           opt    +
    oblique root    up
  sem-struct
    ingest
      agent value ^$var1
      theme value ^$var2
            sem   liquid
    aspect
      phase      end
      iteration   1

```

*Up* in *drink up* may be treated as a derivational morpheme. An inflectional closed-class morpheme—for instance, the marker of verbal tense—may also contribute to aspectual meaning. In combination with the lexical meanings of many verbs (e.g., *lose*, *arrive*, *contribute*, *hide*, *refuse*), the syntactic meaning of simple past tense in English adds the phasal value of BEGIN/CONTINUE/END and the iteration value 1. The progressive tense forms, for those verbs that have them, would contribute the phasal value CONTINUE but they will not provide a clue for the value of the iteration feature.

Aspectual values are contributed to the meaning of a proposition not only through verbs. A number of adverbials denoting time have aspectual meaning as well. Compare *he sat on the bench on Wednesday* and *he sat on the bench every Wednesday*. The aspectual value of the former is PHASE: CONTINUE, ITERATION: 1; that of the latter is PHASE: CONTINUE, ITERATION: MULTIPLE.

```
wednesday-n1
  syn-struct
    root    wednesday
    cat     n
  sem-struct
    time    get-proposition-time
    aspect  iteration  1
```

In the SEM-STRUC above, *get-proposition-time* is the call to a function that returns an absolute value of time that maximally includes the full date, the day of the week and the time of day. The above meaning of *wednesday* captures such usages as *on Wednesday*, *last Wednesday* or *next Wednesday*. We expect to be able to establish rather accurately the time relation between the time when the text is written or read (as can be determined, for example, from the dateline of a newspaper article) and the Wednesday that is the time of the proposition in the sentence. We separate the second nominal meaning of *Wednesday* to account for iterative events that happen on Wednesdays, that is, to capture such usages as *(he goes to the park) every Wednesday*, *on Wednesdays* or simply *Wednesdays*. This meaning is realized using three different syntactic constructions and uses the ontological concept WEDNESDAY, a descendant of TIME-PERIOD.

```
wednesday-n2
  syn-struct
    1    root    wednesday
        cat     n
        mods   root    OR every each
    2    root    wednesday
        cat     n
        number plural
    3    root    on
        cat     prep
        object  root    wednesday
                cat     n
                number  plural
  sem-struct
    1 2 3  wednesday
    aspect
            iteration  multiple
```

We present only the temporal meaning of *every*, which is reflected in the value of the element-type property of the set that is used to represent universal quantification. The syntactic constraints in this entry include a reference to the word that *every* modifies (represented as *\$var1*). It is the meaning of that word that is quantified, that is, is listed as the value of the element type of the set. The filler of the SEM facet of the element type property of the format of set is present to constrain the meaning of that word to temporal units, so that if the input is *every table* instead of *every Wednesday*, this sense of *every* will not be selected.

every-adj2

```
syn-struct
  root   $var1
  cat    n
  mods   root   every
sem-struct
  time
    set1
      element-type  value  ^$var1
                    sem    temporal-unit
      complete      value  yes
```

The multiple value of ITERATION may be contributed by an adverb such as *often*. *Often* modifies a verb (represented as *\$var1*). The meaning of *often* is represented in exactly the same way as the meaning of *many*—the difference between these words is syntactic, as *many* modifies nominals. The meaning of *often* is represented as follows. There is a set, set1, of all possible occurrences of the EVENT marked by *\$var1*. A subset, set2, of this set refers to all the occurrences of this EVENT that are referred to in the input. The property MULTIPLE of this subset represents the relative cardinality of the subset and the entire set in terms of the standard abstract scalar range {0,1} used in ontological semantics. The particular numbers in the lexicon entry represent the meaning of *many* (for comparison, the numbers 0.6-0.9 would represent the meaning of *most*).

often-adv1

```
syn-struct
  root   $var1
  cat    v
  mods   root   often
sem-struct
  set1
    element-type  value  ^$var1
                  sem    event
    complete      value  yes
  set2
    subset-of     value  set1
    multiple      sem    0.33-0.66
  aspect
    iteration     multiple
```

The following two entries describe two of the meanings of *time*. Both meanings are triggered when the word is preceded by a number or a word with the meaning of a number—as in *seven times* or *one time*—which supplies the filler for the aspectual property of ITERATION.

time-n5

```
syn-struct
  root   $var1
  cat    v
  mods   root   time
         cat    n
         number singular
         mods   root   OR one single
```

```

sem-struct
  ^$var1
  aspect
    iteration  1

time-n6
  syn-struct
    root  $var1
    cat   v
    mods  root    time
          cat     n
          number  plural
          mods    root  $var2
          cat     number

sem-struct
  ^$var1
  aspect
    iteration  ^$var2

```

Processing of aspectual values consists of instantiating the meanings of aspect present in the lexical entries for all the input words and unifying them among themselves and with the clues present in the results of syntactic analysis of the input. We posit that the absence of aspectual clues in the lexical entries for the words in the input should lead to the assignment of the aspectual features PHASE: CONTINUE, ITERATION: 1.

### 8.5.2 Proposition Time

Propositions in the ontological-semantic TMR have the property of time—indicated through reference to the start and/or end times of the event which is the head of the proposition. The values of time in this version of the ontological-semantic microtheory of time can be absolute and relative. Absolute times (e.g., June 11, 2000) may be either directly reported in the input, or it might be possible to calculate them based on the knowledge of the time of the speech act in the input sentence using the procedure GET-PROPOSITION-TIME first introduced in the discussion of the lexical entry for *Wednesday* in Section 8.5.1 above.

Speech acts can be either explicit (*IBM announced that it would market applications of voice recognition technology*) or implicit (*IBM will market applications of voice recognition technology*). The time of an explicit speech act is marked on the meaning of the communicative verbs (*announce*, in the example). The time of an implicit speech act must be derived using ellipsis processing—the simplest clue, if available, is the dateline of the article or message containing the statement.

If absolute times cannot be determined, a significant amount of information about temporal relations among the various propositions and speech acts in the input text can still be extracted and represented. In fact, one and the same function can be used for determining the absolute and the relative temporal meanings, with the difference that, in the former case, the values will be actual, though possibly partial (for example, referring only to dates, not times of the day), absolute specifications of times, while relative times, which are partial orderings on times of events in a text, are represented in the TMR using the operators ‘after’ (>), ‘before’ (<) and ‘at’ (=) applied to start

and end points of other events, even if the absolute times of these referent events are unknown. As a shorthand, we allow the '=' operator to apply to time intervals. In such cases, the semantics of the operator is cotemporaneity of those intervals.

A detailed example of calculating propositional temporal meaning at the grain size of dates is given below. This procedure will allow the specification of absolute times if the time of speech is known to the system and relative times otherwise. The function given in the example details how to determine the temporal meaning of the sentence *he will leave on <day-of-the-week>*, where <day-of-the-week> is any of {Monday, ..., Sunday}. The function is described in pseudocode for legibility.

```

get-proposition.time :=
  case day-of-the-week
  monday
    case get-speech-act.time
      tuesday          = speech-act.time.date + 679
      wednesday        = speech-act.time.date + 5
      thursday         = speech-act.time.date + 4
      friday           = speech-act.time.date + 3
      saturday         = speech-act.time.date + 2
      sunday           = speech-act.time.date + 180
      monday           = speech-act.time.date + 781
      undetermined     AND (> speech-act.time.date + 1) (< speech-act.time.date + 7)
  tuesday
    case get-speech-act.time
      tuesday          = speech-act.time.date + 7
      wednesday        = speech-act.time.date + 6
      thursday         = speech-act.time.date + 5
      friday           = speech-act.time.date + 4
      saturday         = speech-act.time.date + 3
      sunday           = speech-act.time.date + 2
      monday           = speech-act.time.date + 1
      undetermined     AND (> speech-act.time.date + 1) (< speech-act.time.date + 7)
  wednesday
    case get-speech-act.time
      tuesday          = speech-act.time.date + 1
      wednesday        = speech-act.time.date + 7
      thursday         = speech-act.time.date + 6
      friday           = speech-act.time.date + 5
      saturday         = speech-act.time.date + 4
      sunday           = speech-act.time.date + 3
      monday           = speech-act.time.date + 2
      undetermined     AND > (speech-act.time.date + 1 < speech-act.time.date + 7)

```

---

79. The number added to the speech act time here and elsewhere in the example stands for the number of days.

80. This input is unlikely to occur. The correct input would be 'tomorrow.'

81. This input is unlikely to occur. If this analysis facility serves a human-computer dialog system, the system should in this state generate a clarification question: "Do you mean today or in a week's time?"

```

thursday
  case get-speech-act.time
    tuesday      = speech-act.time.date + 2
    wednesday    = speech-act.time.date + 1
    thursday     = speech-act.time.date + 7
    friday       = speech-act.time.date + 6
    saturday     = speech-act.time.date + 5
    sunday       = speech-act.time.date + 4
    monday       = speech-act.time.date + 3
    undetermined AND (> speech-act.time.date + 1) (< speech-act.time.date + 7)
friday
  case get-speech-act.time
    tuesday      = speech-act.time.date + 3
    wednesday    = speech-act.time.date + 2
    thursday     = speech-act.time.date + 1
    friday       = speech-act.time.date + 7
    saturday     = speech-act.time.date + 6
    sunday       = speech-act.time.date + 5
    monday       = speech-act.time.date + 4
    undetermined AND (> speech-act.time.date + 1) (< speech-act.time.date + 7)
saturday
  case get-speech-act.time
    tuesday      = speech-act.time.date + 4
    wednesday    = speech-act.time.date + 3
    thursday     = speech-act.time.date + 2
    friday       = speech-act.time.date + 1
    saturday     = speech-act.time.date + 7
    sunday       = speech-act.time.date + 6
    monday       = speech-act.time.date + 5
    undetermined AND (> speech-act.time.date + 1) (< speech-act.time.date + 7)
sunday
  case get-speech-act.time
    tuesday      = speech-act.time.date + 5
    wednesday    = speech-act.time.date + 4
    thursday     = speech-act.time.date + 3
    friday       = speech-act.time.date + 2
    saturday     = speech-act.time.date + 1
    sunday       = speech-act.time.date + 7
    monday       = speech-act.time.date + 6
    undetermined AND (> speech-act.time.date + 1) (< speech-act.time.date + 7)

```

The above function can be extended for treating such sentences as *he left on <day-of-the-week>, he leaves next week / month / year, he returns in <number> minutes / hours / days / weeks / months / years, etc.*

Proposition time is assigned not only when there is an overt lexical reference to time in the input, as in the above examples. In fact, most sentences and clauses in input texts will contain references to times through tense markers on verbs. In such cases, relative time values will be introduced in

the propositions, with time marked with reference to the time of speech. Thus, simple past tense forms will engender time values  $< \text{SPEECH-ACT.TIME}$  in the TIME property of the relevant proposition.

If both a tense marker and an overt lexical time reference are present in the input, the temporal information can be recorded in the TMR multiply, both as an absolute and a relative reference to time filling the TIME property of the same proposition. Usually, they will be in agreement with each other, e.g., a statement issued on June 12, 2000, that the President left for Camp David on June 9, 2000. Occasionally, however, there may be a discrepancy, as in a statement issued on June 12, 2000, which reads as follows, *It may turn out on June 15, 2000, that the President left for an emergency Middle East summit on June 14.* In the case when the temporal meanings clash, the absolute reference gets priority.

While the above examples involve time references to points (or at least are interpreted as such), overt references to time intervals are equally frequent in texts, e.g., *the meeting lasted for five hours* or *the meeting lasted from 10 a.m. till 3 p.m.* In such cases, temporal meanings are encoded using the start and end points of the intervals. Similarly to the case with point references to time, both relative and absolute (or partial absolute) values are acceptable.

### 8.5.3 Modality

Consider the following English verbs: *plan, try, hope, expect, want, intend, doubt, be sure, like (to), mean, need, choose, propose, wish, dread, hate, loathe, love, prefer, deign, disdain, scorn, venture, afford, attempt, contrive, endeavor, fail, manage, neglect, undertake, vow, envisage.* Their meanings have much in common. They all require complements that are infinitival or gerundive constructions (that is, modifying another verb) and their meanings express an attitude on the part of the speaker toward the content of the proposition headed by the meaning of the verb that the verbs from the above list modify. The syntactic similarity of these verbs is not terribly important. Indeed, there are verbs in English (e.g., *help* or *forget*) with the same syntactic behavior but whose meaning is not attitudinal. As is customary in linguistic and philosophical literature, we refer to these attitudinal meanings as modal (cf., e.g., Jespersen 1924: 313—where the term ‘mood’ is used for modality; Lyons 1977: 787-849). Unlike most linguists and philosophers (Fillmore 1968: 23; Lewis 1946: 49; Palmer 1986: 14-15), ontological semantics limits the category of modality to just these attitudinal meanings, having posited ASPECT and TIME as parameters in their own right. The grammatical counterparts of these, the categories of aspect, tense and mood, are treated as clearly distinct from the above semantic categories, though they provide clues for assigning various values of the ontological semantic parameters.

As shown in Section 7.1.1 above, modalities in ontological semantics are represented in the following format:

modality	
type	epistemic   epiteuctic   deontic   volitive   potential   evaluative   saliency
attributed-to	*speaker*
scope	<any TMR element>
value	[0.0, 1.0]
time	time

Modalities can scope over entire propositions, proposition heads, other concept instances or even instances of properties. Note that MODALITY.TIME is often different from PROPOSITION.TIME, as in *I was sure they would win*, said about yesterday’s game.

**Epistemic** modality expresses the attitude of the speaker toward the factivity of the proposition in the scope of the modality. As Lyons (1977:793) correctly points out about epistemic modality, “there is some discrepancy ... between the sense in which philosophers employ the term and the sense in which it has come to be used in linguistic semantics.” While “epistemic logic deals with the logical structure of statements which assert or imply that a particular proposition, or set of propositions, is known or believed,” epistemic modality in ontological semantics measures the degree of certainty with regard to the meaning of a proposition on the part of the speaker.

The values of epistemic modality range from “The speaker does not believe that X” (value 0) through “The speaker believes that possibly X” (value 0.6) to “The speaker believes that X” (value 1). In what follows we present examples of the use of epistemic modality in TMR fragments for actual texts.

Nomura Shoken announced that it has tied up with Credit 109.

modality-2

type	epistemic
attributed-to	corporation-11
scope	merge-6
time	< speech-act.time
value	1.0

For every proposition in TMR there will be an epistemic modality scoping over it. When there are no overt clues for the value of this modality, that is, when a statement is seemingly made without any reference to the beliefs of the speaker (as, in fact, most statements are), then it is assumed that the value of the epistemic modality is 1.0. There may be additional epistemic modalities scoping over parts of the proposition, as mentioned above. For example, in the TMR for the sentence below, two epistemic modalities are captured. The first modality is practically a default value. It simply says that somebody actually made the assertion and there are no clues to the effect that this could not have happened. The second modality is more informative and says that the amount of investment given in the input sentence is only estimated and not known for a fact, and we record this by assigning the value of the modality at 0.8-0.9. If the word *guessed* were used instead of *estimated*, the value would go down to 0.3-0.7.

The amount of investment in the joint venture is estimated at 34 million dollars.

modality-5

type	epistemic
attributed-to	*speaker*
scope	invest-43
value	1.0
time	< speech-act.time

modality-6	
type	epistemic
attributed-to	*speaker*
scope	invest-43.theme
value	0.8-0.9
time	< speech-act.time

Epistemic modality is the device of choice in ontological semantics for representing negation:

The energy conservation bill did not gain a sufficient number of votes in the Senate.

modality-7	
type	epistemic
attributed-to	*speaker*
scope	make-law-33
value	0.0
time	< speech-act.time

**Epiteuctic**<sup>82</sup> modality scopes over events and refers to the degree of success in attaining the results of the event in its scope. The values of epiteuctic modality range from complete failure with no effort expended as in *they never bothered to register to vote* (value 0) to partial success in *they failed to recognize the tell-tale signs of an economic downturn* (value 0.2-0.8) to near success in *he almost broke the world record in pole vaulting* (value 0.9) to complete success in *they reached the North Pole* (value 1.0).

Epiteucticity may be seen as bearing some resemblance to the notion of telicity. In standard examples (Comrie 1976: 44-45), “[s]ituations like that described by *make a chair* are called telic, those like that described by *sing* atelic. The telic nature of a situation can often be tested in the following way: if a sentence referring to this situation in a form with imperfective meaning (such as the English Progressive) implies the sentence referring to the same situation in a form with perfect meaning (such as the English Perfect), then the situation is atelic; otherwise, it is telic. Thus from *John is singing* one can deduce *John has sung*, but from *John is making a chair* one cannot deduce *John has made a chair*. Thus a telic situation is one that involves a process that leads up to a well-defined terminal point, beyond which the process cannot continue.”

We have several serious problems with telicity. First, is it a property of the meaning of a verb or is it not? *Sing* is atelic but *sing a song* is telic. Worse still, *making a chair* is telic but *making chairs* is atelic. More likely, it is the situation described by a text rather than the semantic property of a verb that can be telic or atelic. Recognizing this, Comrie remarks that “provided an appropriate context is provided, many sentences that would otherwise be taken to describe atelic situations can be given a telic interpretation.” However, we cannot accept Comrie’s final positive note about telicity: “although it is difficult to find sentences that are unambiguously telic or atelic, this does not affect the general semantic distinction made between telic and atelic situations.” The reason for that is that texts in natural languages are not normally ambiguous with regard to telicity. As

---

82. From the Classical Greek for ‘success.’

ontological semantics is descriptive in nature, it has a mandate to represent the intended meaning of input texts. If even people cannot judge the telicity of most inputs but are still able to understand the sentences correctly, then one starts to suspect that the category of telicity is spurious: it does not contribute any useful heuristics for successful representation of text meaning.<sup>83</sup>

We also have problems with Comrie's test. It works well in English. It does not seem to "translate" well into other languages, such as, for instance, Russian. The Russian equivalent of the English Progressive for *pet* 'sing' is *poju* '(I) sing' or '(I) am singing.' The equivalent of the English Perfect is *spel* 'have sung,' and it is not implied by *poju*. To complicate matters even further, the difference between Russian perfective and imperfective verbs referring to the same basic event is derivational, and therefore lexical rather than inflectional and therefore grammatical. In fact, we suspect that it is the neatness of the above English test that suggested the introduction of the concept of telicity in the first place. As we argued in Section 4.2 above (see also Section 3.5.2), there is no isomorphism between syntactic and semantic distinctions, so we are not surprised that telicity is hard to pin down semantically.<sup>84</sup>

Epiteucticity also resembles Vendler's (1967) accomplishment and achievement *aktionsarten*. Vendler associates accomplishments with durative events and achievements with punctual ones. We have found a use for this distinction in ontological semantics, and epiteucticity seems to cover both these *aktionsarten*. Ontological semantics also easily accommodates the phenomena that gave rise to the discussions of telicity. The content of fillers of the EFFECT property of events in the ontology describes the consequences and results of the successful completion of events. Interestingly, some of these events would be characterized as atelic. For example, one of the effects of the event BUILD is the existence of the THEME of this event; one of the effects of SLEEP, clearly an atelic event, is that the PATIENT of SLEEP is refreshed and is not sleepy anymore.

Unlike telicity, epiteucticity passes the procedural test in ontological semantics—we need this modality to account for the meanings of such English words as *fail*, *neglect*, *omit*, *try*, *attempt*, *succeed*, *attain*, *accomplish*, *achieve* as well as *almost*, *nearly*, *practically* (cf. Defrise's 1989 in-depth analysis of the meaning of the French *presque*).

**Deontic** modality in ontological semantics deals with the semantics of obligation and permission. "Deontic modality," Lyons (1977: 823) writes, "is concerned with the necessity or possibility of

---

83. Another example of a widely promulgated distinction in linguistic theory that we have shown to be devoid of utility for ontological semantics is the dichotomy between attributive and predicative syntactic constructions for adjectives (see Raskin and Nirenburg 1995). Categories like these make us wonder whether the litmus test for introducing a theoretical construct should not be its utility for language processing. In other words, in our work, we oppose introducing distinctions for the sole reason that they can be introduced if this does not help resolve any problems in automatic analysis and synthesis of natural language texts. Theoretical linguistics does not follow this formulation of the Occam's razor principle.

84. In more recent literature, the term 'telic' was reintroduced by Pustejovsky (1995) as the "purpose and function," an "essential aspect of a word's meaning." The examples of English nominal meanings that include the property of telicity show that this property is similar to the lexical function Oper of Meaning-Text theory (e.g., Mel'čuk 1974), essentially meaning "the typical operation performed with an object." These examples do not make the nature of the telic/atelic dichotomy clear simply because they do not make use of any such distinction, at least not in Comrie's terms.

acts performed by morally responsible agents” (see Section 1.1 above). This modality is used to express the speaker’s view that the agent of the event described in the proposition within the scope of a deontic modality statement is either permitted to carry out the event or is actually under an obligation to do so.

The scale of deontic modality measures the amount of free will in the actions of an agent: unconstrained free will means zero obligation or maximum permissiveness; rigid obligation means absence of free will. The polarity of the scale does not matter much. Ontological semantics defines 0.0 as the value for the situations of unconstrained free will, while the other extreme (value 1.0) of the scale corresponds to the situations of absence of free will, or unequivocal obligation. The values of deonticity in the examples below range from no obligation whatsoever in (53) (value 0.0), to some hint of a non-binding obligation in (54) (value 0.2) to the possibility of an obligation in (55) (value 0.8) to an absolute obligation in (56) (value 1.0).

- (53) British Petroleum may purchase crude oil from any supplier.
- (54) There is no stipulation in the contract that Disney must pay access fees to cable providers.
- (55) Kawasaki Steel may have to sell its South American subsidiary.
- (56) Microsoft must appeal the decision within 15 days.

To give but one example, the modality for (56) will be recorded as follows:

```

modality-9
  type           deontic
  attributed-to  *speaker*
  scope         appeal-6
  value         1.0
  time          > speech-act.time

```

Ontological semantics analyzes negative deonticity as in *I do not have to go to Turkey* as a zero epistemic modality scoped over the deontic modality value of 1.0 (deduced from the lexical clue *have to* in the input).

**Volitive** modality expresses the degree of desirability of an event. Among the English words that provide lexical clues for volitivity are: *want, hope, plan, wish, desire, strive, look forward to, be interested in*, etc. The scale of the volitive modality corresponds to the intensity of the desire. For example, in *also angling for a solid share in the Philippine rolled steel market is Nissho Iwai Corp.*, the volitive modality value is as follows:

```

modality-19
  type           volitive
  attributed-to  *speaker*
  scope         acquire-8
  value         > 0.7
  time          > speech-act.time

```

**Potential** modality deals with meanings that describe the ability of the agent to perform an action.

These meanings are carried by modal verbs such as *can* and *could*, as well as other lexical clues, such as *be capable of*, *be able to*, etc. The scale of the potential modality goes from “Action is not doable by Agent” (value 0) through “Action is definitely doable by Agent” (value 1.0). For example, in *less than 90% of California’s power demand can be met by in-state utilities* the value of the potential modality is as follows:

modality-21	
type	potential
attributed-to	*speaker*
scope	provide-67
value	1.0
time	= speech-act.time

**Evaluative** modality expresses attitudes to events, objects and properties. One can also evaluate another modality. Evaluation goes from the worst, from the speaker’s point of view (value 0.0) to the best (value 1.0). English lexical clues evoking evaluative modality include such verbs as *like*, *admire*, *appreciate*, *praise*, *criticize*, *dislike*, *hate*, *denigrate*, etc. as well as such adjectives as *good* or *bad*. As we have shown elsewhere (Raskin and Nirenburg 1995, 1998), such adjectives provide one of the clearest examples of syntactic modification being distinct from semantic modification: the meanings of these adjectives express evaluative modality and do not modify the meaning of the nouns they modify syntactically. The meanings of *John said that he liked the book he had finished yesterday* and *John said that he had finished a good book yesterday* are identical and contain the following element:

modality-23	
type	evaluative
attributed-to	*speaker*
scope	book-3
value	> 0.7
time	< speech-act.time

**Saliency** modality expresses the importance that the speaker attaches to a component of text meaning. Unlike most of the other modalities, saliency does not usually scope over an entire proposition. This is made manifest in the paucity of verbal clues for saliency scoping over propositions. Indeed, this list seems to be restricted to constructions in which *important*, *unimportant* and their synonyms introduce clauses, e.g., *It is unimportant that she is often late for work*, where a low value of saliency scopes over *she is often late for work*. There are many more cases in which saliency scopes over objects, as manifested by dozens of adjectives with meanings synonymous or antonymous to *important*.

Ontological semantics also uses saliency to mark the focus / presupposition (or topic / comment, or given / new, or theme / rheme) distinction (see Section 3.6 above). In the sentence *the man came into the room*, *the man* is considered the given and *came into the room*, the new. In the sentence *a man came into the room* the given and the new are reversed. English articles, thus, provide lexical clues for the given / new distinction. Not every sentence is as easy to analyze in terms of the given / new distinction. Some sentence elements cannot be categorized as either given or new,

e.g., *works as* in *my father works as a teacher*. While *my father* and *a teacher* may change places as given and new depending on the context, *works as* always remains “neutral.” The most serious difficulty with recognition and representation of this distinction is, however, its contextual dependence and the complexity and variety of textual clues for it as well as its wandering scope. Indeed, the clues can be present outside the sentence, outside the paragraph and even outside the entire discourse. Clearly, ontological semantics expends a limited amount of resources for the recovery of this distinction, specifically, it relies on those lexical clues that are readily available.

The saliency modality is also used to represent special questions. As we indicated above (see Section 6.7 above), some fillers of TMR properties remain unbound after the analysis of a text—because there was no mention of such property or filler there. For example, the TMR for the phrase *the brick house* will bind the property of MATERIAL but will leave the properties such as SIZE or COLOR of the concept instance of HOUSE unfilled. In order to formulate the question *What color is this house?* we include a saliency modality with a high value scoped over the property of COLOR in the frame for HOUSE. Note that this question may either appear in the text or be posed by the human interlocutor in a human-computer question answering system.

## 8.6 Processing at the Suprapropositional Level

When both the basic semantic dependencies and the proposition-level microtheories have been processed, it is time to take care of those properties of the text that scope over multiple propositions, possibly, over the entire text. In the present implementation of ontological semantics, we have identified the following microtheories at this level: reference, discourse and style. The comparatively tentative tone of the sections that follow reflects reality: in spite of many attempts and a number of proposals, the state of the art offers little reliable knowledge on these phenomena and few generally applicable processing techniques for them. The current implementations of ontological semantics do not include fully blown microtheories of reference, discourse and style, either. We do believe, however, that ontological semantics enhances the chances for these phenomena to be adequately treated computationally. This hope is predicated on the fact that no other approach benefits from overt specification of lexical and compositional meaning as clues for determining the values for these phenomena.

### 8.6.1 Reference and Co-Reference

The creation of a TMR is a proximate goal of text analysis in ontological semantics. The TMRs contain instances of ontological concepts—events and objects. These instances may be mentioned for the first time in the sum total of texts processed by an ontological semantic processor. Alternatively, they can refer to instances that have already been mentioned before.

In the discussion that follows we assume that the particular ontological semantic system opts to retain the knowledge accumulated during its operation, and we expect most of the systems to follow this route. In this regard, ontological semantics seems to be the first semantic theory that understands the importance of retaining knowledge for accurate meaning representation. In general, it is fair to say that descriptive linguistics is not interested in the actual usages of linguistic expressions, limiting itself to their potential rather than realized meanings. It is hard to imagine in linguistic literature a situation where the description of the sentence *The cat is black* includes any information about the identity of the cat, those of the speaker and the hearers, the time and place

and other parameters of the actual utterance.

Specific utterances of linguistic expressions have never been in the center of linguistic interest even though studying the use of the definite and indefinite articles in English and of the equivalent devices in other languages (see Raskin 1980) calls for the introduction of the notion of instantiation. Bally's (1950) venture into 'articulation,' his term for instantiation, is a rare exception. The use of object instances would provide a much better explanation of determiner usage than those offered in literature, most of it prescriptive and, therefore, marginal to linguistics. The philosophy of language (see, e.g., Lewis 1972) has attempted to accommodate instantiation by indexing such arguments as speaker, hearer, time, place, etc. in the propositional function. And while the difference between variables and indexed constants has seeped into formal semantics (see Section 3.5.1 above), no actual descriptions have been produced, as neither the philosophy of language nor formal semantics are interested in implementing linguistic descriptions.

Instantiation is, of course, very much in the purview of natural language processing. It is precisely because ontological semantics deals both with standard linguistic descriptions that never refer to instances and the description of specific utterances that it claims standing in both theoretical and computational linguistics.

If an unattested instance appears in a text, a knowledge-retaining ontological semantic processing system would store it in the Fact DB, giving it a new unique identifier. When an instance has already been mentioned before, it is appropriate to co-index a new mention of the same concept instance with the previous mentions of the same instance. The former process establishes reference, the latter, co-reference.

We define co-reference as identity of two or more instances of ontological concepts appearing in TMRs. Instantiation in ontological semantics is the device for expressing the phenomenon of reference. Thus, for us, co-reference is a kind of reference. References to instances of objects and events can be made using such expressive means as:

- direct reference by name, as in *Last week Bill Clinton went on an official visit to Turkey, Greece and Kosovo*;
- pronominalization and other deictic phenomena, as in *The goal of his visit to these countries was to strengthen their ties with the United States*;
- indefinite and definite descriptions of various kinds, as in *This was the President's first trip to the Eastern Mediterranean*.
- ellipsis, as in *He traveled [to Turkey, Greece and Kosovo - elided] by Air Force One*;
- non-literal language (that is, metaphors, metonymies and other tropes), as in *The White House (metonymy) hope that the visit will stem the tide (metaphor) of anti-American protests in Greece*.

The literature on co-reference (Hobbs 1979, Aone and Bennett 1995, Shelton 1997, Baldwin 1997, Azzam *et al.* 1998, Mitkov 2000) tends to focus centrally on objects, usually realized in language as noun phrases. We extend the bounds of the phenomenon of co-reference to event instances. In the current format of the TMR, objects and events are the only independently instantiated ontological entities. Therefore, in our approach, co-reference can exist only among independent instances of ontological concepts and can be defined also as reference to the same

concept instance, which entails the identity of all properties and their values. Identical attribute values introduced by reference (as in *My street is as broad as yours*) are represented by direct inclusion of the actual value, in this case, street width, in the TMR for both streets. At the same time, the techniques that languages use to introduce co-reference and, therefore, the processing techniques with regard to co-reference, are also used for marking reference of this and other kinds. These techniques are based on economical devices that natural language has for establishing property values in one concept instance by saying that they are the same as those in another. This is not reference proper if by reference we understand a relationship between language expressions and instances of ontological events or objects. Here we have a relationship between language expressions and properties of ontological instances.

For instance, in (57), *then* refers to June 1985, therefore the time of Mary not knowing the fact that John was thinking of leaving the army is also set to June 1985. What this means is that the value of the time property for the first event, John's thinking about leaving the Army, is mentioned directly, in absolute terms (see Section 8.5.2 above); the time property for the second event, Mary's not knowing this, gets the same value by virtue of a correct interpretation of the meaning of *then*.

(57) In June 1985, John was already thinking of leaving the Army, and Mary did not know it then.

Examples in (58) and (59) illustrate how the same mechanism works for other parametric properties—aspect and modality. Both sentences introduce two event instances, for one of which the values of modality and aspect are established directly, while for the other, the same values are recorded through a correct interpretation of the meaning of *so did*.

(58) Every Wednesday Eric sat in the park, and so did Terry.

(59) Brian wanted to become a pilot, and so did his brother.

Processing reference involves first identifying all the potentially referring expressions in textual inputs. This is carried out in ontological semantics by the basic semantic dependency builder (see Section 8.2 above) which, when successful, generates all the object and event instances licensed by the input. The next step is to decide for each instance whether it appears in the input text for the first time or whether it has already been mentioned in it. The final result of this process is establishing the chains of co-reference relations within a single text.

Next, for each co-reference chain or single reference found in the text we need to establish whether the ontological semantic system already knows about this instance, that is, whether it is already listed in the nascent TMR or the Fact DB. If the latter contains the appropriate instance, the information in the input text is used to update the knowledge about that instance: for example, if the TMR or the Fact DB already contains information about Eric from (X1) then we will only need to add the knowledge about his park visiting habits—unless that information is already listed there. If no such instance exists, it is created for the first time. In general, as schematically illustrated in Figure 20, the content of the Fact DB is used, together with that in the nascent TMR, as background world knowledge in routine semantic analysis, that is, the previously recorded information is made available to the analyzer, the inference maker or the generator when they process a new mention of the same ontological instance. It is noteworthy, however, that in practical imple-

mentations of ontological semantics, information is recorded in the Fact DB selectively, to suit the needs of the application at hand (see Section 9.4 below).

The processing of reference relies on a variety of triggers and clues. The most obvious triggers in natural language are pronouns, certain determiners, and other indexical expressions (Bar Hillel 1954, Lewis 1972). Once such a trigger is found in a text, a text-level procedure for reference resolution is called. Less obviously, any language expression that refers to an event or object instance triggers the text-level reference resolution procedure. As usual, ontological semantics includes available clue systems in its microtheory of reference resolution, e.g., the numerous heuristics proposed for resolving deixis, anaphora and cataphora in natural languages (Partee 1984b, Reinhart 1983, Webber 1991, Fillmore 1997, Nunberg 1993, Mitkov and Boguraev 1997). Most of these proposals cannot use semantic information. Most systems and approaches simply disregard semantics and base their clues on morphological and syntactic properties (e.g., matching grammatical gender between a personal pronoun and a noun casts a vote for their co-referentiality). Those approaches that include semantics in their theoretical frameworks uniformly lack any descriptive coverage for developing realistic semantic clues for reference resolution.

What triggers the Fact DB-level reference resolution procedure is the set of single references and co-reference chains established as a result of text-level reference resolution. The clues for determining co-reference here include matching or congruency values of all ontological properties. For example, if a Fact DB entry says about John Smith that he resides at 123 Main St. in a certain town and the new text introduces an instance of John Smith at the same address, this state of affairs licenses co-reference. Co-reference may be established not only by exact matching but also by subsumption: if the instance in the Fact DB says about John Smith that he is between 60 and 75 years of age while the instance obtained from a new text says that he is between 65 and 70, this difference will not necessarily lead to refusing co-reference.

Database-level reference-related operations involve not only resolution but also inference-making routines typically used when a system (e.g., a question answering or an MT system) seeks additional information about a fact, specifically, information that was not necessarily present in an input text (in the case of question answering, the text of a query). Such information may be needed for several purposes, for example, to find an answer to a question or to fill an information extraction template or to find the most appropriate way to refer to an entity in the text that a system is generating. For example, the Fact DB stores many event instances in which a particular object instance participates, so that if a system seeks a definite description to refer to George W. Bush, it might find the fact that he won the 2000 Presidential election, and generate the definite description “the winner of the 2000 election.”

In the TMR of a text, reference is represented as a set of co-reference chains found in this text by the reference resolution routine. Each such chain consists of one (reference) or more (co-reference) concept instances. The instances in a chain may come either from the same proposition or, more frequently, from different propositions. It is because of the latter fact that reference and co-reference phenomena have been assigned to the suprapositional level (see Section 6.4 above).

### **8.6.2 TMR Time**

This suprapositional parameter is organized similarly to reference in the sense that it also con-

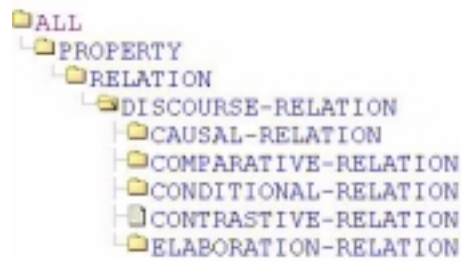
tains sequences of proposition-level values. While in the case of co-reference, each chain establishes identity of its links, each chain in TMR time states a partial temporal ordering of proposition-level time values. In the literature on processing temporal expressions (Allen 1984, Allen and Hayes 1987, Shoham 1987, Gabbay *et al.* 1994, 2000) less attention has been paid to TMR time than to proposition time. Moreover, this literature typically does not focus on any discovery procedures or heuristics for extracting time values from text, concentrating instead on the formalism for representation of those values, once they are determined. Establishing and representing partial temporal orderings is a complex task, as usually there are few explicit clues in texts about relative order of events.

The process of determining TMR time takes a set of proposition-level times as input and attempts to put all of them on a time axis or at least order them temporally relative to each other, if none of the time references is absolute. As it is not expected that an absolute ordering of time references is attainable—texts typically do not specify such an absolute ordering, as it is seldom critical to text understanding—the output may take one of two forms. For those chains that include absolute time references, an attempt would be made to place them on a time axis, so that the result of TMR-level time processing will be a set of time axes, with several time references marked on each. Alternatively, if a connected sequence of time references does not include a single absolute time reference, the output takes the form of a relative time chain.

No chain can contain two temporal references for which the temporal ordering cannot be established. For example, consider the following text: *Pete watched television and Bill went for a walk before they met in the pub.* Three event instances will be generated by the semantic analyzer. The proposition-level time microtheory will establish two temporal relations stating that the meeting occurred after Pete watched TV and that it occurred also after Bill went for a walk. There is no way of determining the relative temporal ordering of Pete's and Bill's individual actions. Therefore, the TMR time microtheory will yield two partial temporal ordering chains, not one.

### **8.6.3 Discourse Relations**

Discourse relations are also a suprapropositional phenomenon. However, they are treated and represented in an entirely different way from reference. Unlike reference, discourse relations are ontological concepts. They form a subtree of the RELATION tree in the PROPERTY branch in the ontology. The incomplete but representative set of discourse relations in Figure XX, with their properties specified, has been developed by Lynn Carlson at NMSU CRL in the framework of the discourse relation microtheory within ontological semantics (see also Carlson and Nirenburg 1990 and Nirenburg and Defrise 1993 for earlier versions).



**Figure 39. The top level of the subtree of discourse relations in the CAMBIO/TIDES implementation of the ontology**

The approach of ontological semantics to discourse analysis differs from that taken by current and recent research in this field (Grosz and Sidner 1986; Mann and Thompson 1988; Webber 1991, Marcu 1999). That research, by necessity, establishes discourse relations over elements of text—sentences and clauses; in ontological semantics the fillers for the domain and range of discourse relations are TMR propositions. Like all the other approaches, however, in defining and using discourse relations, ontological semantics seeks to establish connectivity over an entire text by connecting meanings of individual propositions with the help of discourse relations.

Discourse relations in a text are established using both textual and conceptual clues. Like all the approaches to discourse analysis, the current ontological semantic microtheory of discourse analysis uses all the well-known lexical and grammatical clues. Lexically, it is done using the meanings of words like the English *so*, *finally*, *therefore*, *anyway*, *however*, most prepositions ranging over clauses (e.g., *After John finished breakfast he drove off to work*). and others. Grammatically, the clues can be found, for instance, in the relative tense and aspect forms of verbs in the matrix and a subordinate clause: *Having finished breakfast, John drove off to work*. Ontological semantics adds the opportunity to use conceptual expectation clues. If, for example, two or more propositions are recognized as components in the same complex event stored in the ontology, then, even if the overt textual clues are missing, the discourse analysis module will establish discourse relations among such propositions based on the background world knowledge from the ontology or the Fact DB, in the case when the corresponding complex event was already instantiated and recorded there. Additional discourse analysis clues are provided by the co-reference chains in the TMR.

It is well-known both in theoretical and computational discourse analysis that the current state of the art fails to supply comprehensive and definitive solutions for the problem. Specifically for the purposes of developing computational applications, there are much too few reliable and broadly applicable discovery procedures for establishing discourse relation values. While the blame for that may be assigned by some to the lack of trying, we believe that the asemantic approaches are inherently doomed to fail in supplying the necessary results. We hope that the addition of conceptual clues will facilitate progress in discourse analysis.

#### 8.6.4 Style

Style is a suprapositional parameter that is given a value through the application of a function over the values of specific propositional parameters. In other words, the style of an entire text is calculated on the basis of style values generated for individual propositions. Just as in the case of discourse analysis, the clues for establishing style may be textual or conceptual, with only the former familiar from literature on stylistics (e.g., DiMarco and Hirst 1988, Hovy 1988, Tannen 1980, Laurian 1986). With respect to textual clues, the literature on text attribution (e.g., Somers 1998) contains methods that can be helpful for determining the values of style properties as defined in Section XX above. These methods tend to operate with the help of a predefined limited set of clues (or a small set of statistical regularities to watch), not systematically connected with the lexicon. In ontological semantics, however, the stylistic zone of the lexicon provides blanket coverage of constituent stylistic values that are supplied as arguments to the style computation function. The stylistic zone of the lexicon was present in the Mikrokosmos implementation of ontological semantics but did not make it into the CAMBIO/CREST one—only because in neither implementation, the application did not call for a procedure that used the knowledge in that zone. Note that grammatical information contributing to determination of style values, from such obvious phenomena as the length and complexity of sentences to the more subtle case of the persistent use of passive voice in a text that signifies a higher level of formality than the use of active voice, can be used both in asemantic and ontological semantic approaches.