# A Compositional Semantics for Focusing Subjuncts

Daniel C. Lyons

Technical Report CSRI-234
August 1989

Computer Systems Research Institute
University of Toronto
Toronto, Canada
M5S 1A4

The Computer Systems Research Institute (CSRI) is an interdisciplinary group formed to conduct research and development relevant to computer systems and their application. It is an Institute within the Faculty of Applied Science and Engineering, and the Faculty of Arts and Science, at the University of Toronto, and is supported in part by the Natural Sciences and Engineering Research Council of Canada.

# A Compositional Semantics for Focusing Subjuncts

by

Daniel C. Lyons

Department of Computer Science
University of Toronto
Toronto, Ontario, Canada
August 1989

A Thesis submitted in conformity with the requirements
for the degree of Master of Science in the
University of Toronto

# Abstract

**Focusing subjuncts** such as *only*, *even*, and *also* are a class of sentence adverbials that poses a challenge to a compositional semantics. The relationship between the focusing subjunct and its **focus**, an item that it draws attention to in the sentence and that is usually accented, is particularly problematic, as are the presuppositional aspects of the meaning conveyed by focusing subjuncts. The meaning of focusing subjuncts has in the past been expressed in first-order or higher-order logics, which are computationally impractical representations of meaning.

This research overcomes these problems by introducing a semantics for focusing subjuncts that is compositional, computationally practical, able to distinguish between asserted and non-asserted meaning, sensitive to intonation, and cross-categorial. The semantics are made compositional and computationally practical by adapting the system of Mats Rooth, which employs a higher-order logic. A frame-based meaning representation is used instead. This is facilitated by the use of an relatively unconstrained syntactic treatment of focusing subjuncts, and by capturing their meaning with two operators on sentential semantic forms. A two-part semantic representation along the lines proposed by Lauri Karttunen and Stanley Peters allows non-asserted aspects of meaning to be captured.

The semantics are incorporated into the program IDEO, a semantic interpreter that uses a GPSG grammar modified to account for focusing subjuncts.

## Acknowledgments

I would like to thank my supervisor Graeme Hirst for the time, effort, and suggestions he has given me over the past year and a half, and my supervisor Diane Horton for her very significant help with this thesis. I am indebted to Brendan Gillon for his thoughtful comments, which he provided on quite short notice. I would also like to thank Barb Brunson for pointing me at some of the most useful literature on focusing subjuncts and for her advice on linguistic points. I am grateful to Chuck Pilkington for his invaluable production assitance, to Mark Ryan for beating a path for me to follow these past months, and to Carolyn Cho for her constant encouragement.

I would like to acknowledge my financial benefactor, the Natural Sciences and Engineering Research Council, whose support made possible the completion of this research.

Finally, I thank my family for their moral support and encouragement during my past seven years of university. Thanks, Mom, Dad, and Kathy.

# Contents

# List of Figures

# Chapter 1

# Introduction

## 1.1 Focusing subjuncts

**Focusing subjuncts** such as *only*, *even*, and *also* are a subclass of the sentence-element class of adverbials, according to the grammar of Quirk et al. (1985).[1] Focusing subjuncts draw attention to a part of a sentence, called the **focus** of the focusing subjunct, which often represents 'new' information. They are usually realized by adverbs, but occasionally by prepositional phrases. *Simply* and *in particular* are focusing subjuncts of the following sentences (Quirk et al., 1985, §8.116):

(1) I was simply ⟨taking my dog for a walk⟩.[2]

(2) ⟨The workers⟩, in particular, are dissatisfied with the government.

Focusing subjuncts fill the semantic role of **modality**: they emphasize, approximate, or restrict their foci (see appendix A). They modify the force or truth value of a sentence, especially with respect to its applicability to the focused item. In this thesis, only the adverb realizations of focusing subjuncts are considered, and these lexical items themselves will henceforth be referred to as focusing subjuncts. The purist takes the term to refer exclusively to a grammatical function or sentence role, rather than a part of speech.

Quirk et al. (1985, §8.116) divide focusing subjuncts into several categories. **Restrictives** indicate that the utterance is true, either **exclusively** or **predominantly**, of the item focused. Examples of the former kind, called **exclusives**, include *just* and *only*. Examples of the latter kind, called **particularizers**, are *especially* and *mainly*. **Additives**, such as *also*, *even*, and *too*, indicate that the utterance is *additionally* true of the part focused.[3]

[1] Appendix A summarizes Quirk's definition and taxonomy of adverbials.

[2] Note: In the example sentences throughout this thesis, words in small capitals have intonational stress, and angle brackets ⟨⟩ enclose the *focus* of a focusing subjunct. Unacceptable sentences are preceded by an asterisk (*), and sentences that are self-contradictory are preceded by a hash mark (#). In some of the example sentences in Chapter 1, the focusing subjunct (or candidate) is italicized, and square brackets [ ] are used to set off the constituent to which the focusing subjunct adjoins.

[3] A listing of the focusing subjuncts is included in appendix A.

Focusing subjuncts are interesting and difficult to deal with, for several reasons.

- The words that act as focusing subjuncts are polysemous. In some of their senses they play a completely different role.
- Focusing subjuncts can adjoin to many different types of constituent. They can occur at almost any syntactic position in a sentence.
- Focusing subjuncts can precede or follow the item that they focus (draw attention to), and need not be adjacent to this item.
- They may focus almost any constituent.
- Because of the above three points, it is difficult to determine the intended syntactic argument (adjunct) and focus of a focusing subjunct.
- The location of intonational stress has an important effect on the meaning of a sentence containing a focusing subjunct.
- Focusing subjuncts don't fit into the simple slot–filler semantics that seem adequate for handling many other sentence elements (see Section 1.2.3). At best, their semantic effect is to transform the semantic representation of the constituent they modify in some predictable compositional way (Hirst, 1987, p. 72).
- They carry a lot of pragmatic "baggage". It is necessary to resort to talking about presuppositions and conversational and conventional implicatures (see Section 1.5 below) in order to fully explain the meaning of a focusing subjunct.

The first six of these problems are elaborated upon below. The last two are addressed in Section 1.3.

**Polysemy** The different senses of *only* in (3) are obviously semantically very close. Yet only (3.1) contains *only* in the sense in which it is a focusing subjunct. In (3.2), it is a conjunction, and in (3.3), *only* acts as an intensifier, weakening the force of the verb *joking*.

**(3)**
1. She is charming *only* [to her wealthy clients].
2. He could have done it, *only* he didn't make an effort.
3. I was *only* joking.

Many words that can act as focusing subjuncts also have senses in which they do not. These polysemous words include *just*, *also*, *merely*, *too*, and *rather*. Some of the different senses of *too* and *just* are given in (4) and (5), respectively. Many of the sample sentences are from Quirk et al. (1985).

(4) *Uses of* too*:*

1. I brought the wine and the bread *too.* [a conjunct]
2. John has seen it *too.* [an additive focusing subjunct]
3. *Too* true. [an emphasizer subjunct]
4. You did *too.* [as in (4.3), as an asseverative]
5. I'm not *too* sure about that. [an intensifier subjunct]
6. *A*: He's clever. *B*: *Too* right, he is. [as in (4.5), in Australian English]

(5) *Uses of* just*:*

1. I *just* can't understand it. [an emphasizer subjunct]
2. You can get a B grade *just* for that answer.
   [an exclusive restrictive focusing subjunct]
3. *Just* exactly what do you expect?
   [as in (5.2), modifying another focusing subjunct, *exactly*]
4. She'll *just* be out for a few minutes. [an intensifier subjunct]
5. When I saw her, she had *just* come back. [a time-relationship item subjunct]

**Syntactic position** Focusing subjuncts can occupy almost any position in a sentence.

(6)
1. *Even* Bob was there. [sentence initial]
2. I *simply* was taking my dog for a walk. [before auxiliary]
3. I had *simply* been taking my dog for a walk. [after first auxiliary]
4. I could have *simply* been taking my dog for a walk.
   [between second and third auxiliaries]
5. I have been *simply* taking my dog for a walk. [before main verb]
6. She has invited *as well* some of her own family. [after verb]
7. We didn't buy any beer, *either.* [sentence final]

It appears that focusing subjuncts can adjoin to any maximal projection (an observation due to Barbara Brunson, personal communication):

(7)
1. John introduced only [$_{NP}$ ⟨BILL⟩] to Sue. [NP]
2. Ross even [$_{VP}$ washed ⟨the DOG⟩]. [VP]
3. John has seen it [$_{PP}$ ⟨near his back DOOR⟩] even. [PP]
4. I'm old but I'm also [$_{AP}$ ⟨HAPPY⟩]. [AP]
5. Even [$_{S'}$ if the agreement were SIGNED], it wouldn't be binding. [S′]

**Position relative to their focus** Focusing subjuncts are flexible with respect to their position, so much so that multiple positions are possible even for a given focus (Quirk et al., 1985, §8.119):

**(8)** I've noticed the fox in my garden and...

1. John has *even* seen it ⟨near his back DOOR⟩.
2. John has seen it *even* ⟨near his back DOOR⟩.
3. John has seen it ⟨near his back DOOR⟩ *even*.

**Promiscuity of focus** Focusing subjuncts can focus the subject, the object, a main or auxiliary verb, a complement, another adverbial, or an adjective phrase.

**(9)**

1. *At least* ⟨ten workers⟩ reported sick yesterday. [subject]
2. His father wouldn't give the money *nor* would he ⟨LEND⟩ it. [main verb]
3. The old man *simply* ⟨COULD⟩ not have done the deed.
   [(modal) auxiliary verb]
4. I was *simply* ⟨taking my dog for a walk⟩. [predication]
5. I gave the dog a bone, and I gave the ⟨cat⟩ one *as well*. [indirect object]
6. He became happy, and he *also* became ⟨RICH⟩. [complement]
7. I will be away ⟨tomorrow⟩ *as well*. [adverbial]
8. She is charming *only* to her ⟨WEALTHY⟩ clients. [adjective phrase]

**Ambiguity of focus** Consider the following sentence:

**(10)** John could also see his wife from the doorway.

Depending on where the stress falls when the sentence is spoken, *also* has the choice of focusing the main verb, some part of the predication, or the whole predication (Quirk et al., 1985, §8.117) ((11.3) and (11.5) were noticed by Diane Horton, private communication).

**(11)**

1. ⟨JOHN⟩ could also see his wife from the doorway.
   [so could Tom]
2. John could also ⟨SEE⟩ his wife from the doorway.
   [as well as being able to talk to her]
3. John could also see ⟨HIS⟩ wife from the doorway.
   [as well as Tom's wife]
4. John could also see ⟨his WIFE⟩ from the doorway.
   [as well as her brother]
5. John could also ⟨see his WIFE⟩ from the doorway.
   [as well as do other things from the doorway]

6. John could also see his wife ⟨from the DOORway⟩.
   [as well as from further inside the room]

7. John could also ⟨see his wife from the DOORway⟩.
   [he could also do other things]

**Importance of intonation** The sentence 10 is ambiguous among all of the interpretations in 11 in written text, but not in spoken language. Intonation effects partly disambiguate the sentence. For example, only the readings (11.4) and (11.5) are possible if the word *wife* is stressed intonationally. However, intonation alone does not indicate which of these two senses is intended.

## 1.2 Compositional semantics

Frege's Principle of Compositionality is often cited as a desirable feature of a system of semantics (Dowty, Wall and Peters, 1981, p. 8). As we shall see in Section 1.3, focusing subjuncts present a challenge to a compositional semantics. In this section, I define what makes a semantics compositional, and explain the sense in which compositionality can be considered a matter of degree. Two compositional semantics are presented, Montague's (1973) PTQ system and Hirst's (1987) Absity.

The following two properties characterize a compositional semantics:

(12)
1. Each word and well-formed syntactic phrase is represented by a distinct semantic object.
2. The semantic representation of a syntactic phrase is a systematic function of the representation of its constituent words and/or phrases.

Note that this definition assumes a syntax in which phrases are complex objects constructed from constituents according to some rules (phrase-structure rules). In a compositional semantics, the syntax drives the semantics. To each syntactic phrase construction rule there corresponds a semantic rule that specifies how the semantic objects of the constituents are (systematically) combined or **composed** to obtain a semantic object for the phrase.

It is the word *systematically* in the definition that makes compositionality a scale rather than an absolute property. The systematicity of a function is a nebulous, gray concept. My intuition is that the more a function's value depends on a set of arguments and not on external variables or constants, the more it is a systematic function of these arguments. As well, the simpler the function, the more systematic it is.

There are several alternatives to compositional semantics (Hirst, 1987, p. 27). The first alternative is that the meaning of a phrase depends on the situation in which the sentence it occurs in is uttered. Therefore, it is not just a systematic function of its *constituents'* meaning. To the extent that the meaning of English sentences depends upon pragmatics (situation and context), any truly accurate semantics for English will not be purely compositional.

A second alternative is that the relationship between the meaning of a phrase and its parts is just not systematic. For instance, in Woods's (1967) *procedural semantics*, whole clauses are matched against a store of word patterns, each associated with a procedure call. The meaning of a sentence is identified with the procedure call corresponding to the pattern it matches. In this semantics, a word or phrase may have meanings which vary arbitrarily from one pattern to another, and hence from sentence to sentence. Thus, per Hirst (1987, p. 27), "The meaning of an individual word varies idiosyncratically with the other words in the same sentence."

### 1.2.1 Advantages of compositionality

The motivation behind Frege's articulation of the idea of compositional semantics was that natural language itself is for the most part compositional.

- People compose phrases and words whose meaning they understand in order to predict the meaning of sentences they've never seen before. They do not have to memorize idiosyncratic senses for every sentence they can comprehend.
- Natural language also appears to have the characteristic that each well-formed constituent has semantic significance. People understand the "meaning" of phrases such as *with the new shampoo* and *the dog* independently of a sentential context. Furthermore, words and phrases retain, for the most part, their meaning from sentence to sentence. They do not vary idiosyncratically depending on the context.
- People don't construct a complete syntactic parse of a sentence before beginning to decipher its meaning.

Certain computational benefits accrue to semantic interpretation with the use of a compositional semantics.

- If the semantics is formulated correctly, the semantic representation attributed to a constituent can guide the parsing and disambiguation of what follows it in a sentence. For instance, syntactically possible prepositional phrase attachments can be ruled out on semantic grounds if the preceding noun phrase has been interpreted.
- Rather than storing and looking up idiosyncratic meanings for each sentence, the semantic interpreter builds sentence meanings recursively from a relatively small store of word and idiom meanings. The interpreter can thus achieve a far wider coverage of the language from a given size of the dictionary of meanings.
- By virtue of being systematic, the algorithm (function) to construct the meaning of constituents is simple and uniform, having fewer special cases, in a compositional semantics.

### 1.2.2 Montague: PTQ

Montague semantics may be the canonical compositional semantics. It was introduced by Richard Montague in his "PTQ" ("Proper Treatment of Quantification") paper (Montague, 1973) in the form of a syntax and semantics for a fragment of English. It has been extended widely in many different and probably incompatible ways by other researchers, including Mats Rooth, whose extensions of PTQ to treat *even* and *only* will be outlined in section 2.5.

The foundation of Montague's semantics is a formal mathematical **model** of the world. The semantics are **model-theoretic**: the meaning of a sentence is a function from possible worlds, or **indexes**, to truth values. Equivalently, the meaning is the set of possible worlds in which the sentence is true. Semantic objects in Montague semantics are model-theoretic objects such as individuals in the world model, truth values, individual concepts (or intensions, which are functions from indexes to individuals). Other types of object such as properties of individual concepts (which are sets of individual concepts) are recursively defined on top of these base types. In Montague semantics, expressions of **intensional logic** (IL) are used as an intermediate representation between English and the semantic objects. English expressions are translated into IL expressions, which can be interpreted to determine the model-theoretic objects that they **denote.**

The PTQ grammar consists of pairs of rules. The first is a syntactic rule and the second is a **translation** (semantic) rule that specifies an IL translation for the English phrase constructed by the first. Hirst calls this property **tandem** operation of the semantic and syntactic rules. The semantic rules make use of Montague's theory of types of semantic objects. Each type of syntactic constituent corresponds to one of the types of IL. Each semantic rule combines objects of an expected type.

A common kind of translation rule is the **functional application** rule, so named because it forms the semantic object of a phrase by *applying* the translation of one of its parts as a function of that of another. A typical rule of this kind has this form:

**(13)** If $X$ is of type $\alpha$ and $Y$ is of the type of functions from intensions of $\alpha$ to $\beta$, then $Z$ is $Y({}^\wedge X)$ and has type $\beta$.

(${}^\wedge$is the intensional operator). The semantic and syntactic rules and types are paired such that there is never a mismatch in the type of semantic objects the semantic rule expects. For example, the translation rule T4 (Dowty, Wall and Peters, 1981, p. 192), shown below in (14), dictates that *John*$'({}^\wedge$*walks*$')$ be the translation for the phrase *John walks*, which the corresponding syntactic rule S4 (also shown in (14)) gives as the value of $F_4(\alpha, \delta)$. *John*$'$ and *walks*$'$ are the translations of *John* and walks, respectively.

**(14)** S4: If $\alpha \in P_T$ and $\delta \in P_{IV}$, then $F_4(\alpha, \delta) \in P_t$, where $F_4(\alpha, \delta) = \alpha\delta'$, and $\delta'$ is the result of replacing the first *verb* in $\delta$ by its third person singular present form.

T4: If $\alpha \in P_T$ and $\delta \in P_{IV}$, and $\alpha, \delta$ translate into $\alpha', \delta'$ respectively, then $F_4(\alpha, \delta)$ translates into $\alpha'(^\wedge\delta')$.

A second kind of rule introduces words syncategorematically. These words are not assigned a semantic (or syntactic) type. They are specified explicitly by name rather than by type in the syntactic and semantic rules. The method by which syncategorematic introduction rules construct a semantic object is usually more complicated than functional application. One of the simpler rule pairs (S11a, T11a) introduces the word *and* (Dowty, Wall and Peters, 1981, p. 198):

(15) S11a: If $\phi, \psi \in P_t$, then $F_8(\phi, \psi) \in P_t$, where $F_8(\phi, \psi) = \phi$ and $\psi$.

T11a: If $\phi, \psi \in P_t$, and $\phi, \psi$ translate into $\phi', \psi'$ respectively, then $F_8(\phi, \psi)$ translates into $[\phi' \wedge \psi']$.

A third and final kind of semantic rule translates most of the words in the grammar.

Montague semantics is nothing if not compositional. Strong typing allows the single device of functional application to be used to construct the semantic objects of most phrases. Syncategorematic rules make the mapping to semantic objects of phrases from those of their parts just a little less systematic.

Montague semantics is not frequently implemented in AI systems because of two drawbacks. The first is that it is computationally impractical. Its semantic objects are infinite objects and huge sets of things such as functions, possible worlds, and functions among these things. The second is that the meaning it attributes to a sentence is the set of possible worlds in which the sentence is true. In AI, "we are interested not so much in whether a state of affairs is or could be true in some possible world, but rather in the state of affairs itself" (Hirst, 1987, p. 32). An additional drawback of truth-conditional semantics in general is that they are hard to extend to sentences that are not declarative.

### 1.2.3 Hirst: Absity

Absity is a semantic interpreter, a program that maps text in a natural language to a representation of its meaning. Absity uses a highly compositional semantics for English that the author says is "Montague-inspired" (Hirst, 1987, p. 33). Like PTQ, it makes use of strong typing of semantic objects and of the operation of syntax and semantics in tandem (Hirst, 1987, p. 44).

Absity's underlying representation of the world is a knowledge base consisting of frames, rather than a model. A **frame** is a collection of stereotypical knowledge about some topic or concept (Hirst, 1987, p. 12). A frame is usually stored as a named structure having associated with it a set of **slots** or roles that may be assigned **values** or **fillers**. Absity's semantic objects belong to the types in a frame representation language called Frail (Charniak, 1981; Wong, 1981a, 1981b), rather than the types of a model theory. Absity uses the following types of semantic object:

| Syntactic type | Semantic type |
|---|---|
| Sentence | Frame statement |
| Proper noun | Frame statement |
| Pronoun | Frame statement |
| Common noun | Frame |
| Determiner | Frame determiner |
| Noun phrase | Frame statement |
| Preposition | Slot name |
| Prepositional phrase | Slot–filler pair |
| Subordinate clause | Slot–filler pair |
| Verb | Frame |
| Adverb | Slot–filler pair |
| Auxiliary | Slot–filler pair |
| Verb phrase | Frame descriptor |
| Clause end punctuation | Frame determiner |

Figure 1.1: Absity type correspondences

- a frame name
- a slot name
- a frame determiner
- a slot–filler pair
- a frame description (frame with zero or more slot–filler pairs)
- *either* an instance *or* frame statement (atom *or* frame determiner with frame description)

A frame determiner is a function that retrieves frames or adds them to the knowledge base. A frame description describes a frame in the knowledge base. The filler of a slot is either an atom, or it is an instance, specified by a frame statement, of a frame in the knowledge base. The relationship between syntactic categories and semantic types is shown in Figure 1.1 (Hirst, 1987, p. 48).

The semantics of Absity are quite compositional. There is a very small set of methods of composing objects. The method used depends entirely on the type of the objects:

- A slot and a frame statement produce a slot–filler pair with the frame statement as the filler.
- A slot–filler pair and a frame name produce a frame description in which the slot of the slot–filler pair belongs to the named frame.
- A slot–filler pair and a frame description produce another frame description with the slot–filler pair further modifying the frame description.

```
(a ?u (wash ?u
   (agent=(the ?x
              (person ?x (propername=Ross))))
   (patient=(the ?y (dog ?y)))
   (instrument=(a ?z (shampoo ?z (age=new))))
))
```

Figure 1.2: An Absity frame statement

- A frame determiner and a frame description produce a frame statement in which the description is the argument of the frame determiner.

Other combinations of semantic objects are not allowed. Thus, the computation of the semantic object of a phrase from those of its parts is very systematic.

Absity's semantic strong typing and compositional systematicity have significant consequences for parsing. The major one relates to the handling of subject and object noun phrases. Note that a verb and a noun phrase have types that cannot be combined, since a verb is a frame and a noun phrase is a frame statement. Absity overcomes this problem by treating object and subject noun phrases as prepositional phrases, which have a type (slot–filler pair) than can be combined with a verb's type (frame name). This is accomplished by inserting case flags called *pseudo-prepositions* into the parse tree adjacent to these noun phrases. These case flags represent the important semantic role of syntactic position. They are processed by the semantics just as if they were prepositions: they belong to the type *slot*.

The notation Hirst uses for frames is illustrated in Figure 1.2, which is a frame statement translation of the sentence

**(16)** Ross washed the dog with a new shampoo.

## 1.3 The semantics of focusing subjuncts

**Semantic non-compositionality** It is a non-trivial task to devise a compositional semantics of focusing subjuncts, for a number of reasons. First, consider the sentence

**(17)** Ross even washed ⟨the DOG⟩.

Semantically, it seems most natural that *even* take its focus, *the dog*, as its functional argument. However, the argument available to *even* is its syntactic sister (the verb phrase [$_{VP}$ washed the DOG] in the case of (17)). To make the focus of *even* available as an argument requires that *even* or its focus be moved during semantic interpretation. Adherence to the compositionality principle is complicated by the non-adjacency of the focusing subjunct to its focus. The semantics of focusing subjuncts that this thesis proposes overcomes this problem by allowing a focusing subjunct to take its syntactic sister as its syntactic argument.

Secondly, even assuming the availability of the focus as an argument, it is not clear what a focusing subjunct actually means. Consider sentence 18, which means, roughly "Ross washed the dog and he also washed something else and he was not expected to wash the dog."

(18) Ross washed even ⟨the DOG⟩.

The sentence without *even* might be assigned the semantic representation

```
(19) (wash  tense=past  agent=Ross  patient=dog)
```

It would be inadequate to represent sentence (18) as just

```
(20) (wash  tense=past  agent=Ross  patient=(even  argument=dog))
```

since this does not capture the idea that Ross cleaned something else. More appropriate seems the representation

```
(21) (and
       (wash  tense=past  agent=Ross  patient=dog)
       (wash  tense=past  agent=Ross  patient=?X)
     )
```

which is to be read, roughly, as "Ross washed the dog and he also washed something else." An approach of this type will be adopted in this thesis.

Thirdly, the semantic function that is assigned to a focusing subjunct must be capable of accepting arguments of different types, since both the focus of a focusing subjunct and its syntactic argument (its sister node) vary in type. But a usual feature of compositional semantics is that logical and linguistic arguments of a phrase or lexical item must be of a specified type.

**Pragmatic considerations** According to Fraser (1971), the sentence (22) would have the three-part interpretation given in (23).

(22) Even ⟨ROSS⟩ washed the dog.

(23)
1. Ross washed the dog.
2. Other people washed the dog.
3. The speaker would not expect Ross to wash the dog.

However, these three parts of the meaning of (22) are not equal in the following respect. For a listener to accept (22) as (truth-conditionally) true, only (23.1) need hold. If (23.1) holds but (23.2) or (23.3) do not, the listener is likely to accept the sentence as true, but with a reservation such as "Well, yes, Ross washed the dog, but no one else did" or "Well, yes, he did wash the dog, but that is just as one should expect" (Karttunen and Peters, 1979, p. 12). In such cases the sentence is often taken to be strictly speaking true, though perhaps inappropriate in most circumstances, on contextual grounds. Therefore, it can be argued that *even* does not affect the truth conditions of the sentence. That is, (22) is true exactly when (23.1) is. For this reason, the sentence is sometimes said to *assert* (23.1) and to *presuppose* (23.2) and (23.3) (see Section 1.5).

Representing the content of (23.3) requires that the speaker's beliefs be represented. Distinguishing between the contributions of (23.1) and those of (23.2) and (23.3) to the meaning of the sentence (22) requires the ability to distinguish defeasible, or "non-asserted", or "presuppositional" meanings from a sentence's "assertion". These concerns fall within the scope of the field called **pragmatics** (Levinson, 1983, pp. 5, 12).

## 1.4 Goals of this research

The goal of this thesis is a semantics for focusing subjuncts that is

- compositional,
- computationally practical,
- able to differentiate between asserted and non-asserted meaning,
- sensitive to intonation, and
- cross-categorial.

The semantics should be compositional for reasons given in Section 1.2.1. The semantics I develop is based on Rooth's (1985) compositional semantics for *even* and *only*, described in Section 2.5 below. Being a compositional semantics, it is set within a formal theory of grammar. In order to make my semantics computationally practical, I set it within an Absity-like frame-based semantic framework: the higher-order intensional logic used by Rooth is intractable. To provide evidence that the semantics is computational practical, it is incorporated into a computational semantic interpreter.

The semantics should differentiate between presupposed (or implicated) and asserted meaning of focusing subjuncts. The semantics should be sensitive to intonation because the meaning of focusing subjuncts depend so much on intonational stress. It should be cross-categorial because focusing subjuncts can occur next to such a variety of syntactic arguments: the semantics must be capable of handling focusing subjuncts in many different constructions using a single definition or lexical entry for each focusing subjunct.

A secondary purpose of this thesis is to illustrate the usefulness of focus (see Section 1.5) and two-part semantic representations in a semantic interpreter. A further secondary purpose is to show that it is possible to avoid the use of higher-order intensional logic, which is computationally impractical and intractable, in capturing the meaning of focusing subjuncts. That is, it is possible to replace the intensional logic with a frame-based representation of meaning in which intensions are not expressed without losing any aspect of the meaning of focusing subjuncts themselves.

This work can be seen as a development of Rooth's and McCord's semantics for focusing subjuncts (see Sections 2.5 and 2.6) within a frame-based semantic framework (Absity-style frames, see Sections 3.2 and 1.2.3) and within a formal theory of grammar (namely GPSG, see Section 3.3).

### 1.4.1 Simplifying assumptions

A completely thorough computational treatment of focusing subjuncts appears impossible given what linguists and AI researchers know now. Linguists have not provided a definitive syntactic explanation of focusing subjuncts and of their association with focus (see Section 2.3). The phenomenon of presupposition is not well-enough understood, though it seems to involve the beliefs of speakers (see *Presupposition* in Section 1.5, below). The representation of beliefs is a problem in AI that is only partially solved. Unsolved AI problems include the use of negative information to make inferences and the maintenance of the consistency of knowledge bases, both of which enter into the semantics of focusing subjuncts (see Section 5.4.2). The solution of these problems is beyond the scope of this thesis.

Instead, the present work is a contribution towards the continuing effort to provide a full account of focusing subjuncts. It ties together some of the previous research towards this effort and extends it in several directions. Like previous research in the field, this thesis makes certain simplifying assumptions with respect to focusing subjuncts, and then undertakes to develop a semantics that is adequate modulo these assumptions. Later research is expected to extend the account of focusing subjuncts to an even more general case in which some of the simplifying assumptions are discharged.

## 1.5 Some definitions

It will be useful to define some of the linguistic and semantic concepts used in studying focusing subjuncts. The following sentence will be used to illustrate these examples:

**(24)** Only Muriel voted for Hubert.

**Presupposition** The groundwork for the analysis of focusing subjuncts was laid by Horn (1969) (see Section 2.2). He decomposes the meaning of (24) into a presupposition (25.1) and an assertion (25.2).

**(25)** 1. Muriel voted for Hubert.

2. No one other than Muriel voted for Hubert.

Horn employs Austin's (1963) definition of a presupposition as an implication of a sentence that is preserved under negation of the sentence. For example, Austin considers the sentence

(26) to be anomalous because its second clause violates the presupposition of the first—namely, that John has children—and therefore the first clause has no truth value.

(26) * All John's children are bald, but John has no children.

Horn also provides a definition of presupposition that he says is equivalent, due to Katz and Postal (1964): "a condition that the asker of a question assumes will be accepted by anyone who tries to answer it."

According to Levinson (1983, Chapter 4), presuppositions are characterized by

- their defeasibility in certain contexts,
- their links to particular aspects of surface structure, and
- their behavior in complex sentences: a complex sentence does not inherit intact all the presuppositions of sentences embedded in it.

A more recent account of presupposition is presented by Horton (1988), who shows that constancy under negation is not a necessary property of presuppositions. Presuppositions have also been described as preconditions for the use of certain expressions.

No single definition of presupposition is generally accepted, and none is embraced in this paper. For our purposes, it will be sufficient to note that inferences like (25.1) will be represented separately from the more "obvious" truth-conditions of sentences. The usefulness of making this distinction is broached in Section 3.1.3. I will label as **non-asserted meaning** the propositions that Horn calls presuppositions, such as (25.1), while remaining neutral on the issue of whether they are in fact presuppositions. The label **asserted meaning** will refer to the main assertion or truth condition of a sentence modified by a focusing subjunct.

**Conventional implicature** At least one author (Karttunen and Peters, 1979) (see Section 2.4) has described (25.1) as a **conventional implicature** of (24). According to Levinson (1983, pp. 127–128), this is a category invented by Grice (1975) whose existence is the subject of some debate. Grice used the term "implicature" to include all pragmatic inferences discernible from an utterance, in contrast to "what is *said* or expressed by the truth conditions of expressions". Grice intended "conventional implicature" to mean any non–truth-conditional inference *not* derived from pragmatic principles such as the Gricean maxims, but rather attached by convention to lexical items or expressions. Therefore, they are attached to lexical items rather than the truth-conditions of a sentence, and may be calculated just from the sentence without using pragmatic principles or the context. Conventional implicatures are not cancelable. An example of a conventional implicature provided by Grice is that introduced by the word *but*. This word has the same truth-conditional meaning as the conjunction *and*, but additionally conventionally implicates that there is some contrast between the conjuncts.

Levinson notes that the existence of the category is challenged on the grounds that its only use is to deny that truth-conditional semantics captures the meaning of words and expressions. According to the challengers, among whom Levinson singles out Kempson (1975), alleged cases are either truth-conditional entailments, *conversational* implicatures, which *are* derived from pragmatic principles, or presuppositions. As noted above, I remain neutral as to the status of inferences such as (25.1).

**Focus** For the purposes of this thesis, the concept of focus as a linguistic feature is relevant. Focus is a binary feature, similar to (say) gender and number, which is either absent or present on every constituent at surface structure. The presence of focus can be signalled overtly using various syntactic and phonetic means, but the most common realization of focus is by phonological stress in the focused constituent. The precise meaning of the focus feature is a subject of discussion in this thesis (see Section 3.1.7).

In this thesis, the term *focus* by itself will refer to the focus feature. *Focus of a focusing subjunct* is a separate concept defined below.

**Focus and scope of focusing subjuncts** In his work, Horn describes *only* (when modifying an NP) as a predicate taking two arguments, "the term [$x$] within its scope" and "some proposition [$Fx$] containing that term" (Horn, 1969, p. 99). The meaning of the predicate is then to *presuppose* that the proposition $F$ is true of $x$, and to *assert* that $x$ is the *unique* term of which $F$ is true. For example, in (24), taking $x$ to be *Muriel* and $F$ to be the property of voting for Hubert produces the presupposition (25.1) and the assertion (25.2).

In a similar vein, in Karttunen and Peters's analysis of *even* (see Section 2.4), the "implicature" (my *non-asserted meaning*) of *even* is said to depend on two arguments, a *focus* and a *scope* (Karttunen and Peters, 1979, p. 25). For example, they attribute the implicature (27.2) to sentence (27.1), taking *Mary* as the focus and *Bill likes $x$* as the scope of *even*.

**(27)** 1. Bill likes even MARY.

2. Bill likes other people besides Mary.

In general they describe the implicature contributed by *even* as

**(28)** There are other $x$ under consideration besides $a$ [the focus] such that ...$x$...[the scope].

The scope is the open sentence ...$x$....

The two arguments of focusing subjuncts described by Horn and by Karttunen and Peters are concepts that are referred to in all of the literature on focusing subjuncts. Henceforth, the terminology of Karttunen and Peters will be used to describe these arguments. The term **focus of a focusing subjunct** will refer to the item that the focusing subjunct draws attention to, or *focuses.* This is the same as Horn's argument $x$ in his formulation of the meaning of *only.* The term **scope of the focusing subjunct** will refer to the open sentence in Karttunen and Peters's description of *even*, or, equivalently, the argument $F$ in Horn's description of *only.* This open sentence can always be obtained by removing the focusing subjunct from a sentence, and then replacing the item serving as the focus of the focusing subjunct with a variable (see Section 3.1.1).

# Chapter 2

# Previous research

## 2.1 Overview of the literature

The scant attention that has been paid to the focusing subjuncts in the linguistics and computational linguistics literature is devoted mainly to the adverbs *only* and *even*. In this chapter, we review five different accounts of the semantics of these words:

- Horn (1969) establishes foundations that are on the whole accepted by the later writers—namely, his presupposition/assertion analysis of *only* and his use of the focus and scope arguments described in Section 1.5.

- Jackendoff's (1972) theory of *association with focus* explains the association of *even* (in particular) with its focus as a phenomenon of surface, rather than underlying, structure. It is challenged by Fraser (1971), and later by Rooth (1985).

- Rooth's dissertation (1985) provides a revised account of association with focus, and a semantic theory for the focusing subjuncts *even* and *only*.

- One of the well-accepted claims Horn makes about *even* and *only* is that they introduce non-asserted meaning into a sentence. Karttunen and Peters (1979) propose a multipart model-theoretic semantics to deal with this type of meaning, particularly the conventional implicatures that they maintain are introduced by *even* and *only*.

- McCord's article (1986) is the only one treating focusing subjuncts that the present author was able to discover in the computational linguistics literature. McCord describes a semantic interpreter in which a number of the focusing subjuncts are given semantics that he says follow Horn (1969) and Bergmann (1981).

In addition, several researchers have contributed work that is not reviewed in this chapter, but is referred to in the discussion of research issues in Section 3.1. Anderson (1972) and Fraser (1971) present two competing (mainly syntactic) accounts of *even* that bridge the gap between the earlier theories and Rooth's theory. Bergmann (1981) describes a multi-part semantics for (*inter alia*) *even* and *only*. Gussenhoven (1983) and Ladd (1983) deal with the phenomenon of focus, especially with the relationship between focus and stress (intonation).

## 2.2 Laying the groundwork

Horn (1969) provides what was at the time of its publication a ground-breaking analysis of the meaning of *only* and *even* in terms of presupposition and assertion. Horn employs the following definition of presupposition, which he credits to Austin (1963):

(29) Let $S$ and $S'$ be sentences.

1. If $(S \rightarrow S')$ and $(\neg S \rightarrow S')$, then $S$ *presupposes* $S'$.
2. If $(S \rightarrow S')$ and $(\neg S' \rightarrow \neg S)$, then $S$ *entails* $S'$.
   ($S \rightarrow S'$ is to be read 'From $S$ we can conclude $S'$...')

That is, *presuppositions* are distinguished from *entailments* by their invariance under negation.[1]

Horn considers the semantic contribution of *only* and *even* in a limited number of constructions. Specifically, he looks at *only* modifying an NP or a VP, and *even* when it modifies an NP. Horn describes *only*, when focusing an NP, as a two-place *predicate* (though it is not truth-valued) taking as arguments "the term within its scope" (the focus of *only*), and "some proposition containing that term" (Horn, 1969, p. 99):

(30) Only $(x, F)$[2]
P[resupposes]: $Fx$
A[sserts]: $\neg(\exists y)(y \neq x \ \& \ Fy)$

That is, a sentence with *only* presupposes some proposition, or open sentence, $Fx$, constructed from the sentence, and asserts that this proposition is not true of any item *other* than the focus, $x$ ($x$ is a parameter, not a variable). For example, the sentences

(31)
1. Only Lucifer pities himself.
2. Only Lucifer pities Lucifer.
3. Lucifer pities only himself.

share the presupposition

[1] Horn does not address the question of exactly how $\neg S$ is constructed; yet sentence negation is not a trivial problem. Furthermore, recent work suggests invariance under negation is a default rather than a rule for presuppositions. For a recent account of presupposition that tackles these problems, see Horton (1988).

[2] The actual expression given by Horn is the somewhat confusing

Only $(x = a, Fx)$

Horn does not explain the purpose or meaning of the symbol $a$.

(32) Lucifer pities Lucifer.

but assert, respectively,

(33) 1. No one other than Lucifer pities himself.
2. No one other than Lucifer pities Lucifer.
3. Lucifer pities no one other than Lucifer.

with the appropriate substitutions for the focus $x$ and scope $F$ being, respectively,

(34) 1. $x \leftarrow$ Lucifer and $Fy \leftarrow$ pities($y$, $y$)
2. $x \leftarrow$ Lucifer and $Fy \leftarrow$ pities($y$, Lucifer)
3. $x \leftarrow$ Lucifer and $Fy \leftarrow$ pities(Lucifer, $y$)

Note that more recent definitions of presupposition supersede the one used by Horn, and it is controversial whether (32) should be considered a presupposition of the sentences in (31).

Horn recognizes that intonational stress plays some role in determining the focus and open sentence arguments of *only*. However, he does not specify exactly what this role might be.

Horn's scheme is intended to handle nested occurrences of *only*, though he claims that his analysis of *only* does not rely on the assumption that such occurrences are grammatical. He gives the following example (Horn, 1969, p. 100):

(35) Only John eats only rice.

This sentence is said to presuppose

(36) John eats only rice.

which can be in turn be decomposed into a presupposition, *John eats rice*, and an assertion, *John eats nothing but rice*. The assertion of (35) is given as

(37) Nobody but John eats $\left\{\begin{array}{l}\text{only rice.}\\ = \text{nothing but rice.}\end{array}\right.$

Horn provides no explanation for the apparent non-grammaticality of sentences with two occurrences of *even* or *also*, such as this example:

(38) * Even John even eats rice.

Horn uses a different formulation for *only* when it focuses a verb phrase, as in the following sentence:

(39) Muriel only ⟨voted for HUBERT⟩, she didn't do the laundry.

In sentences like this, *only* takes a property as one of its arguments. Therefore, it is a second-order predicate:

(40) Only $(F, Fx)$
P: $Fx$
A: $\neg(\exists G)(G \neq F \,\&\, Gx)$

Horn's formulation for VP-modifying *only* is flawed, at least notationally. It is not rigorous; for example, it is not clear whether $x$ is intended to be a free or bound in the assertion (A) expression of (40). Another flaw shows up when attempting to cast a sentence such as his example

(41) Muriel only ⟨VOTED⟩ for Hubert (she didn't campaign for him).

into the form of (40). This requires the argument substitutions of (42) to be made in order to produce the correct presuppositions and assertion.

(42)
1. $F \leftarrow$ vote (for someone)
2. $x \leftarrow$ Muriel
3. $Fx \leftarrow$ Muriel voted for Hubert

But it is notationally inconsistent to have $F$ be the property of voting for someone, $x$ be Muriel, and $Fx$ *not* be the proposition that Muriel campaigned for *someone* (i.e. not necessarily Hubert). Therefore, Horn's formulation stated as (40) cannot adequately handle a sentence such as (41). It can only handle sentences in which *only* is interpreted as having the entire verb phrase as its focus.

Horn posits the following analysis of *even*:

(43) Even $(x, F)$
P: $(\exists y)(y \neq x \,\&\, Fy)$
A: $Fx$

That is, "*even* (like *also*) asserts what *only* presupposes and presupposes the negation of what *only* asserts" (Horn, 1969, p. 106). Horn credits Bruce Fraser with the observation that the data of (44) (with focus on *Muriel*) are explained by the fact that clefting specifies uniqueness, as does *only*, while *even* and *also*, incompatibly, presuppose non-uniqueness.[3]

(44) It's $\left\{\begin{array}{l}\text{only}\\ \text{*even}\\ \text{*also}\end{array}\right\}$ ⟨Muriel⟩ who voted for Hubert.

## 2.3 Association with focus

According to Jackendoff (1972, pp. 3, 16), the semantic representation of a sentence consists of four parts:

[3] Notice the possibility of the sequence *It's Muriel who did X. It's also Muriel who did ⟨Y⟩. It's even Muriel who did ⟨Z⟩.* In these cases, Muriel is not the focus of *also/even*.

**Functional structure** Function-argument relationships.

**Modal structure** The conditions under which the sentence purports [*sic*] to correspond to real-world situations.

**Table of coreference** The store of all coreferences between NPs.

**Focus and presupposition** Distinction between new and old information. New information is that which is assumed by the speaker not to be shared by him/her and the hearer.

Jackendoff's usage of the terms *focus* and *presupposition* differs from the usage of the terms in this thesis. His "focus" is semantic rather than syntactic. However, Jackendoff's marker *F* (see below) is the same as the linguistic focus feature defined in Section 1.5. Jackendoff's "focus" is just the semantic material associated with the constituent marked *F* (rule (46) below). Jackendoff formally incorporates focus into his theory; he devotes a chapter of his book to it.

Jackendoff presents two different accounts of focus assignment. The **cleft theory** maintains that focus is assigned at deep structure. Sentences containing an emphatic stress are seen as being cleft at deep structure, with the focus being the predicate of the higher clause, and the presupposition being the lower clause (Jackendoff, 1972, pp. 230–231). Several arguments against this account are advanced, including the possibility of foci that cannot be clefted, such as verbs and quantifiers (Jackendoff, 1972, pp. 234):

**(45)** 1. Did Fred HIT Bill?

2. * Was it HIT that Fred (did to) Bill?

Jackendoff adopts an alternative theory, which he attributes to Chomsky. In this theory, focus is assumed to be assigned, at surface structure, to some phrase containing the main intonational stress of the sentence. A semantic marker *F* is assumed to be present in surface structure and associated with some node containing stress (Jackendoff, 1972, p. 242). This is the same as the focus feature defined in Section 1.5 of this thesis. Jackendoff identifies focus with information status. That is, the *F* marker marks the portion of the sentence whose semantic material is new information. Jackendoff's view on this matter is in accord with that of Gussenhoven (1983), who identifies focus with information status, but at odds with Ladd (1983), who argues that it is possible for contextually "given" items to have focus.

Jackendoff postulates a rule, **focus assignment**, to divide a reading into *focus* and *presupposition* on the basis of syntax, including the *F* marker.

**(46)** *Focus assignment rule:*

In a sentence S, with otherwise determined semantic representation SR, the semantic material associated with surface structure nodes dominated by *F* is the

Focus of S. Substitute an appropriate semantic variable $x$ for Focus in SR to form the function $\text{Presupp}_S(x)$. The presupposition of S is then formed as 2.1 and the assertion is 2.2 (Jackendoff, 1972, p. 247).

$$\lambda x\ \text{Presupp}_S(x) \left\{ \begin{array}{l} \left.\begin{array}{l}\text{is a coherent set}\\ \text{is well-defined}\end{array}\right\}\text{in the present discourse} \\ \text{is amenable to discussion} \\ \text{is under discussion} \end{array}\right. \tag{2.1}$$

$$\text{Focus} \in \lambda x\ \text{Presupp}_S(x) \tag{2.2}$$

Jackendoff emphasizes that the "appropriate semantic variable" must be determined semantically, limited by the notion of "possible contrast" with the focus.

For example, the focus assignment rule derives (47.2) as the function $\text{Presupp}_S(x)$ for the sentence (47.1). This yields the presupposition (47.3) and assertion (47.4) (Jackendoff, 1972, pp. 245–247).

(47)
1. John LIKES Bill.
2. the attitude of John toward Bill is $x$
3. $\lambda x$ [the attitude of John toward Bill is $x$] is under discussion
4. *like* $\in \lambda x$ [the attitude of John toward Bill is $x$]

Jackendoff proposes a rule of **association with focus** that applies to focusing subjuncts. The rule states that *even* and "related words" are *associated with focus* by having the focus in their **range** (see below).[4] Focus is assigned at surface structure, not in deep structure, so the association of *even* with its focus occurs at surface structure. In the case of *even*, the justification for association with focus is that *even* associated with a constituent implies unusualness or unexpectedness, and "if there is something unexpected about the constituent, it must be new to the hearer, and hence by definition part of the focus." Jackendoff does not provide a similar explanation for the association of other focusing subjuncts (such as *just* and *only*) with focus.

In Jackendoff's scheme, differences between the ranges for the various focusing adverbs account for their different distributions (Jackendoff, 1972, pp. 249–250).

**Range of *even*** If *even* is directly dominated by a node X, then X and all nodes dominated by X are in the range of *even*.

**Range of *only*** If *only* is directly dominated by a node X, then X and all nodes dominated by X and to the right of *only* are in the range of *only*.

[4] Because Jackendoff's "focus" is just the semantic material associated with the constituent marked $F$, when he talks about having the focus in a range, we may substitute for "focus" the sense defined in Section 1.5 of this thesis.

That is, *only* (and *just*, which has the same range) cannot precede its focus, while *even* can:

**(48)** 1. * ⟨JOHN⟩ only gave his daughter a new bicycle.

2. ⟨JOHN⟩ even gave his daughter a new bicycle.

Jackendoff provides no independent linguistic motivation for the difference in ranges. Furthermore, there is some evidence that his formulation does not capture the distribution of *only*. For example in the sentence

**(49)** I saw his ⟨ELDER⟩ brother only.

the focus of *only* is to its left. However, the idea of "range" does capture some important differences in the distributions of focusing subjuncts. For example, *alone* is notable for having a range exclusively to its left (Quirk et al., 1985, §8.118):

**(50)** * Alone ⟨ten workers⟩ reported sick yesterday.

According to Jackendoff, the rule of association with focus correctly predicts the restriction on occurrences of *even* to one per sentence, *assuming* that "there is only one focus per sentence". However, later in his chapter on focus, Jackendoff accepts that sentences such as (51) have *more* than one focus (Jackendoff, 1972, p. 260):

**(51)** A: Well, what about FRED? What did HE eat?

B: FRED ate the BEANS.

It appears that both *Fred* and *beans* are focused. Jackendoff accounts for multiple foci and multiple emphatic stress by using two kinds of pitch accents, *A* and *B*. An *A* pitch accent is realized phonologically by high stress on some focus syllable, followed by a phrase that concludes with a fall in pitch. *B* pitch accent is similar, but concludes with a rise in pitch (Jackendoff, 1972, p. 259). *A* and *B* pitch accents are present as features on the focus marker *F*. *B* accent occurs on the "variable" whose value is chosen first, for example the one a questioner is asking about. *A* accent occurs on the variable chosen so as to make the sentence true for the value of the other variable. That is, "the *B* accent defines an *independent variable* and the *A* accent a *dependent variable*" (Jackendoff, 1972, p. 262). In the example above, *Fred* receives *B* accent and *beans* receives *A* accent. Jackendoff points out that the difference between independent and dependent variables is similar to "the traditional notions of *topic* and *comment*" (Jackendoff, 1972, p. 262).

Because Jackendoff's analysis of multiple foci requires one focus to depend on the other, it doesn't cover sentences in which there is no basis for selecting one of the foci as being dependent on the other, such as this standard though somewhat unfortunate example:

**(52)** John can't even sell WHISKEY to the INDIANS.

## 2.4 Multi-part semantics

Karttunen and Peters (1979) propose that much of the presuppositional behavior associated with particles like *too*, *also*, *even*, and *only* is really due to **conventional implicatures** (defined in Section 1.5, and below).

Karttunen and Peters claim that *even* in a sentence such as

(53) Even Bill likes Mary.

plays no role in determining its truth conditions, citing Stalnaker (1974) for support. This claim is also consistent with Horn's (1969) assertion-presupposition analysis of *even*. The speaker's "principal commitment" when uttering (53) is said to be the content of the proposition

(54) Bill likes Mary.

which they call the sentence's *expressed* meaning or *denotation* (my asserted meaning). The speaker also commits himself to

(55) 1. Other people besides Bill like Mary.
2. Of the people under consideration, Bill is the least likely to like Mary.

Sentence (53) is in some way wrong if (54) is false. However, Karttunen and Peters argue that the speaker is usually credited with saying "something that is partially correct" if (54) is true but (55.1) or (55.2) is false. As noted in Section 1.3, a response to (53) in such circumstances might be qualified agreement, such as "Well yes, he does like her; but that is just as one should expect" (Karttunen and Peters, 1979, p. 12).

Karttunen and Peters take the preceding to be evidence that the propositions contained in (55) are *conventional implicatures* of sentence (53), distinct from its *denotation*. They are **implicated**, not asserted, because "the truth of what (53) actually *says* depends solely on whether Bill likes Mary." The implicatures are called **conventional** because they are always present in *even* sentences by virtue of common practice. That is, they do not arise from general conversational principles and the context of the utterance, rather "they simply arise from the presence of the word *even*", and they cannot be canceled, say by modifying (53) as in

(56) # Even Bill likes Mary but no one else does.

which is self-contradictory (Karttunen and Peters, 1979, p. 12). Karttunen and Peters comment that "conventional implicatures are not set apart so they can be challenged in a direct way"; therefore Grice's co-operative principle (Grice, 1975) entails that "a sentence ought to be uttered only if it does not conventionally implicate anything that is subject to controversy" (Karttunen and Peters, 1979, p. 14). But this comment could also be applied to all indirectly conveyed aspects of meaning in general.

Karttunen and Peters propose to represent sentence meaning with a two-dimensional logic[5] in which "truth-conditional" and conventionally implicated aspects of meaning are separated. This is necessary, they say, because expressed and implicated meanings of a phrase are inherited differently by complex sentences in which the phrase is embedded.

**(57)** I just noticed that even Bill likes Mary.

This sentence says that the speaker has noticed that Bill likes Mary, but not that he has noticed that other people like Mary or that Bill is the least likely person to do so. That is, in constructing the meaning of (57), *notice* applies only to the proposition "expressed by" the embedded sentence, not to its conventional implicatures. On the other hand, the conventional implicatures of the embedded sentence *are* inherited intact by sentence (57) (Karttunen and Peters, 1979, p. 13).

Karttunen and Peters describe an extension of Montague's PTQ (1973) grammar, in which every (English) phrase is associated with *two* expressions of **intensional logic** (IL). The first is the **extension expression**, which is identical to the (IL) expression produced by the PTQ grammar. It stands for "what logicians would call the denotation of the phrase." The second is the **implicature expression**, signifying "what the phrase conventionally *implicates* (if it is a sentence), what it contributes (if it is smaller than a sentence) to the conventional implicatures of sentences having it as a part" (Karttunen and Peters, 1979, p. 16).

Each lexical entry in their grammar has its associated extension and implicature expressions. As a phrase $\alpha$ is constructed using a syntactic rule, a paired **translation rule** (semantic rule) assigns an extension expression (as in PTQ), $\alpha^e$, and an implicature expression $\alpha^i$ to the derived phrase. These expressions are computed as a function of the extension and implicature expressions of the derived phrase's constituents. Usually, the extension expression is purely a function of the extension expressions of the constituents.

An example of a pair of syntactic and translation rules is (58). It corresponds to syntactic rule 5 in PTQ (Karttunen and Peters, 1979, p. 18).

**(58)** 1. If $\alpha$ is a transitive verb
and $\beta$ is a noun phrase
then $\alpha\overline{\beta}$ is a verb phrase, where $\overline{\beta}$ is the accusative form of $\beta$.

2. *Translation*: $\langle \alpha^e({}^\wedge\beta^e);\ \ \lambda x[\alpha^i(x, {}^\wedge\beta^e)\ \wedge\ \alpha^h(x, {}^\wedge\beta^i)]\rangle$

Here, $^\wedge$ is the intensional operator of IL. $\alpha^h$ is the result of applying a **heritage function**, $h$, to the extension and implicature expressions of $\alpha$. It is a transformation that must be applied to the inherited conventional implicatures of phrases that $\alpha$ takes as arguments.

[5] This is not what Karttunen and Peters call it. But they acknowledge that their system "is very similar in concept to H. Herzberger's (1973) 'two-dimensional logic'."

According to Karttunen and Peters, the use of the heritage function is necessary to account for the implicatures of intensional verbs such as *hope*, *believe*, and *tell*. Consider for example the sentence

(59) It wasn't Bill who tapped Mary's phone.

which presupposes

(60) Mary's phone was tapped.

Now in sentence (61), (59) is embedded as a complement of the verb *hope*.

(61) John [$_{VP}$ hoped that it wasn't Bill who tapped Mary's phone].

Ordinarily, a sentence or phrase inherits all of the conventional implicatures of its constituents. But in this case, according to Karttunen and Peters, (61) does not inherit (59)'s presupposition. Rather, (61) conventionally implicates that John *believed* Mary's phone was tapped. Karttunen and Peters's solution is that the verb phrase in (61) acquires the conventional implicatures of the embedded sentence (59) *after* these are modified by the effects of the transformation $\text{hope}^h$. That is, $\text{hope}^h$ transforms a proposition such as (60) into an ascription of belief in this proposition (Karttunen and Peters, 1979, pp. 20–21).

Karttunen and Peters illustrate the ability of their grammar to handle problem cases such as *even* sentences. They limit their analysis to cases in which *even* precedes an extensionally interpreted noun phrase (interpreted *de re*) on which it focuses. The denotation and *existential implicature* (see below) that they attribute to *even* are similar Horn's (1969) assertion and presupposition, respectively. According to the authors, their analysis of *even* is easily generalized, for example, to account for *even* with an adjective focus, or to explain the semantics of *too*.

The implicature expression for *even* has two parts. First, the **existential implicature** (my presuppositional assertion) contributed by *even* (Karttunen and Peters, 1979, p. 25) is given as

(62) There are other $x$ under consideration besides $a$ such that ... $x$ ...

Here, $a$ stands for the **focus** of *even*. The open sentence ... $x$ ... bound by the existential quantifier *there are* in (62) is what Karttunen and Peters call the **scope** of *even*.[6] Note that the *focus* and *scope* arguments of *even* correspond closely to the arguments Horn uses to describe the meaning of *even*.

For example, in

(63) Even ⟨Bill⟩ likes Mary.

the focus of *even* is *Bill* and the scope is *x likes Mary*. Karttunen and Peters illustrate that focus and scope are both relevant and independent components of the meaning of an *even* sentence by producing the sentence

[6] Their definitions of focus and scope are the ones adopted in this thesis.

(64) It is hard for me to believe that Bill can understand even ⟨*Syntactic Structures*⟩.

which is ambiguous between two readings. In both readings, the book title *Syntactic Structures* is the focus of *even*. On the first reading, the sentence implicates that Bill's understanding is limited, and *even* has the scope (65.1). On the second reading, the implicature is that Bill can understand other books besides *Syntactic Structures*, and the scope is (65.2).

(65) 1. It is hard for me to believe that Bill can understand $x$.

2. Bill can understand $x$.

A second, **scalar implicature** of *even* is as follows:

(66) For all $x$ under consideration besides $a$, the likelihood that ... $x$ ... is greater than the likelihood that ... $a$ ...

Here, ... $a$ ... is the sentence obtained from the scope ... $x$ ... by substituting the focus, $a$, for $x$.

Karttunen and Peters define a special **even rule** that generates *even* in their syntax syncategorematically, meaning that the word itself is not assigned to any syntactic category. This is how PTQ treats particles such as *every, the*, and *not*. They define a meaning postulate for *even* (actually, an IL definition of $^\wedge\text{even}^i$) that introduces (62) and (66) as conventional implicatures of *even*. Since the *even* rule combines a focus noun phrase (Karttunen and Peters use the PTQ notation "T-phrase") and a scope sentence ("t-phrase") to produce a new sentence (t-phrase) with *even*, a phrase like *even Mary* in *Bill likes even Mary* "is not considered to be a constituent phrase" (Karttunen and Peters, 1979, p. 28). Note that *even* is not treated in a manner similar to the way in which adverbs are treated in PTQ. If it were, it would have to be assigned to some syntactic category, say the category of functions from noun phrases to noun phrases.

The semantic effect of the *even* rule is to substitute the focus of *even* for the pronoun (variable) in its scope sentence, producing the extension expression for the *even* sentence. The rule conjoins the existential and scalar implicatures described above with the conventional implicatures contributed in a natural way by the focus and the scope to produce the implicature expression for the new sentence.

## 2.5 Rooth

In his dissertation, Rooth (1985) examines the interaction between focus, which is equated with intonational prominence, and the adverbs *even* and *only*. This is done within the context of a general examination of the semantic effect of focus. Rooth calls this effect ***association with focus***, a term introduced by Jackendoff (1972) (see section 2.3).

Rooth's work is motivated by such evidence as the minimal sentence pair

(67) 1. I only introduced BILL to Sue.

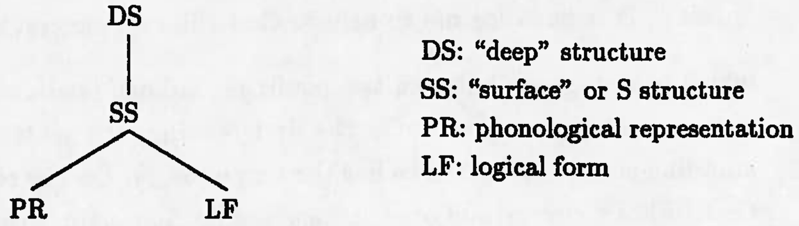

Figure 2.1: The EST organization of grammar

2. I only introduced Bill to SUE.

If the speaker introduces Bill and Tom to Sue, and performs no other introductions, then (67.1) is false and (67.2) is true. Though the two sentences differ only in the location of intonational prominence, they have different truth conditions. This is a case of *only* "associating" with focus (which is on *Bill* in (67.1) and on *Sue* in (67.2)) (Rooth, 1985, pp. 2–3).

Rooth bases his work upon the Extended Standard Theory (EST) organization of grammar (Chomsky and Lasnik, 1977) shown in Figure 2.1. He adopts what he calls the "standard" assumption about focus, that it is "a feature marked on syntactic phrases." It follows from these assumptions that

> focus must be marked in S-structure, since it has correlated phonological/phonetic and semantic/pragmatic aspects; PR is where phonology happens, LF is the locus of semantic interpretation, and SS is the link between the two (Rooth, 1985, p. 10).

That is, Rooth assumes the existence of a focus feature (realized by intonation) on surface syntactic phrases, similar to Jackendoff's $F$ marker. Rooth also assumes *even* and *only* are two cases of the same phenomenon, so that that work on *even* predating his own applies equally well to *only*.

Rooth refers to Selkirk (1984) to justify his linkage of focus with intonational prominence: "The core of Selkirk's proposal is that the phonological reflex of focus is a pitch accent" (Rooth, 1985, p. 18). Another aspect of Selkirk's proposal that Rooth accepts is that a complex phrase can optionally acquire focus by inheriting the feature from one of its daughters: "Focuses can percolate (i) to X-bar phrases from their heads and (ii) to a phrase interpreted as a function plus arguments from the argument phrases" (Rooth, 1985, p. 21).

### 2.5.1 Criticism of earlier theories

Rooth examines two theories of association with focus. He calls one of them the **scope theory**; it is due to Anderson (1972). The other is the standard theory predating Anderson's work and credited to Fischer (1968). Both of these theories predate the Extended Standard Theory, so Rooth takes some liberties in recasting them within an EST framework.

According to the earlier theory, which is essentially the one espoused by Fraser (1971), *even* is adjacent to the focused phrase in deep structure, and is optionally moved to other (surface) positions. Anderson criticized this theory for being unable to accommodate multiple foci, such as occur in (68): *even* cannot be adjacent in deep structure to more than one focused phrase.

(68) John only introduced ⟨BILL⟩ to ⟨SUE⟩.

Anderson proposed (essentially) that *even* "is generated in some single position in underlying structure" (Anderson, 1972, p. 898), which Rooth takes to mean DS, and is later moved into its surface position (in SS). Then, an "interpretive principle", which Rooth takes to mean movement in LF, is applied. This determines which constituent, actually a focused phrase, serves as the focus of *even*. For instance, the surface structure

(69) $[_S$ John $[_{VP}$ only $[_{VP}$ introduced $[_{NP}$ Bill$]_F$ $[_{PP}$ to Sue]]]]

is interpreted by adjoining the focused phrase $[_{NP}$ Bill$]_F$ to *only*, where it is in the appropriate position to be an argument of *only*:

(70) $[_S$ John $[_{VP}$ only $[_{NP}$ Bill$]_{F,2}$ $[_{VP}$ introduced $[_{NP}$ $e]_2$ $[_{PP}$ to Sue]]]].
(the constituents subscripted "2" are co-indexed)

Rooth pokes several holes in Anderson's theory. First, it succumbs to the same criticism as Fischer's regarding multiple foci: only one constituent can be the argument of *only* in logical form. Second, the movement in LF that Anderson requires sometimes violates constraints on movement that have, since Anderson published his paper, been recognized as being applicable to movement in LF as well as to syntactic movement, according to Rooth.

Rooth's third objection to Anderson's theory is that there is no independent principle that forces the appropriate constituent, namely the phrase with focus, to move into an argument relationship with *only*/*even*:

> That focus influences the assertions and conventional implicatures of sentences involving *only*, *even*, and a small group of other adverbs seems to be a marginal fact about English, and we should not have to state a separate principle which covers it (Rooth, 1985, pp. 40–41).

Anderson must posit a restriction that *stipulates* that *only* interact with focus, such as

(71) In LF, *only* must be the sister of a phrase bearing the feature F.

### 2.5.2 Rooth's proposal

Rooth rejects both the standard theory and the scope theory, and puts forth his own **domain selection theory**, which proposes that *only* take the VP adjacent to it in S-structure as its argument. His proposal is not subject to the same criticisms as Anderson's by virtue of not requiring the focused phrase to be an argument of *only*(!) (Rooth, 1985, p. 45). An extension of the theory allows *only* to take arguments other than VPs. Rooth provides a detailed PTQ-style (Montague, 1973) grammar that translates a focused phrase into his Intensional Logic with Focus (ILF).

The semantics attributed to *only* is essentially that of Horn (1969) (see (30) of Section 2.2). However, for simplicity, Rooth drops the non-asserted meaning of *only*. The phrase including the modifier *only* asserts that any property of the form $P(x)$, where $P$ is some *relevant property* derived from the argument VP using focus information, is exactly the property defined by the VP. To illustrate, the meaning of

**(72)** John only [$_{VP}$ introduced BILL to Sue].

is "if John has a property of the form 'introduce $y$ to Sue' then it is the property 'introduce Bill to Sue' " (Rooth, 1985, p. 44, p. 59).

Rooth employs essentially the following logic translation for *only* (his preliminary version):

**(73)** $only'' \;=\; \lambda P \lambda x[\forall Q[[Q\{x\} \;\&\; C(Q)] \;\longrightarrow\; Q=P]]$

Here, $C$ is a free variable representing the "domain of quantification". Specifically, $C$ is "the characteristic function of a set of properties, which we think of as the set of relevant properties." That is, only″ takes a property and an object as arguments, and asserts that the argument property is the unique one that is both true of the object and relevant.

For example, Rooth's theory would produce the translation

**(74)** $\forall P[[P(john) \;\&\; C(P)] \;\longrightarrow\; P = {}^\wedge introduce'(bill, sue)]$

for both of the following sentences:

**(75)**
1. John only introduced BILL to Sue.
2. John only introduced Bill to SUE.

Different truth conditions for the two sentences are obtained by providing different values for $C$. The quantification ($\forall P$) is restricted to properties of the form 'introduce $y$ to Sue' in (75.1), and to relevant properties 'introduce Bill to $y$' in (75.2). The desired values for $C$ are (76.1) and (76.2).

**(76)**
1. $\lambda P \exists y[P = {}^\wedge introduce'(y, sue)]$
2. $\lambda P \exists y[P = {}^\wedge introduce'(bill, y)]$

These result in final translations (77.1) and (77.2) respectively for sentences (75.1) and (75.2):

**(77)**
1. $\forall x[introduce'(john, x, sue) \;\longrightarrow\; x = bill]$
2. $\forall x[introduce'(john, bill, x) \;\longrightarrow\; x = sue]$

The key feature of Rooth's proposal, again, is that the focused phrase is not an argument of *only*. For this reason, (75.1) and (75.2) need not be structurally distinguished in LF. Instead,

> the truth conditional effect of focus ... is a result of a contribution of focus to the selection of domains of quantification (Rooth, 1985, p. 45).

It appears that Rooth's approach involves the use of pragmatic information. The "domains of quantification" can be thought of as pragmatic background against which a sentence is evaluated. Certainly, the function $C$ is related to the presuppositions of a sentence: that the property (76.1) (the property of being introduced to Sue) is "relevant" is a presupposition of (75.1).

Rooth derives a modified version of PTQ in which domains of quantification are incorporated. He lays down some groundwork for his derivation by showing how to construct a **presupposition skeleton** of a sentence with focus. This is just Jackendoff's (1972) function $\text{Presupp}_S(x)$ described in Section 2.3. It is the result of substituting a variable for the focused phrase(s) in a sentence. Rooth's version of the presuppositional skeleton is the **p-set**. The p-set of a sentence $a$ with normal denotation $a'$ is defined as follows (Rooth, 1985, p. 14):

**(78)** *p-set of a*[7]:

1. the set of objects in the model matching $a'$ in type, if $a$ bears the feature $F$;
2. the unit set $a'$, if $a$ is a non-focused non-complex phrase;
3. the set of objects that can be obtained by picking one element from each of the p-sets corresponding to the component phrases of $a$, and applying the semantic rule for $a$ to this sequence of elements, if $a$ is a non-focused complex phrase.

According to Rooth, a critical difference between his definition and Jackendoff's is that the latter's "involves a variable in the position of a focused phrase", whereas Rooth's requires "that a semantic object with variables in the positions of focused phrases [be] available" (Rooth, 1985, p. 16). Rooth notes that a proposal of Chomsky's (1976) is like Jackendoff's in this respect (the variable in Chomsky's proposal being a trace).

Rooth's p-sets are actually used to construct the domains of quantification he requires in his translation rule for *only* (Rooth, 1985, p. 59):

**(79)** *Translation rule for* only*:*

[$_{VP}$ only VP] has the ILF translation:

$$R(C,\ VP,\ \lambda x \forall P[[P(x)\ \&\ P \in C] \longrightarrow P = {}^\wedge VP'])$$

[7] The model referred to in the definition is a formal model à la Montague; $F$ is the focus feature.

The **restriction operator** $R$ produces an expression similar to its third argument, but in this case with the variable $C$ replaced with (essentially) the characteristic function of the p-set of the intension of VP. Rooth explains his preference for this complex formula, in which a variable ($C$) always becomes bound:

> I retained $C$ because I find the possibility that association with focus has something to do with a general phenomenon of selecting domains of quantification interesting (Rooth, 1985, p. 59).

Continuing with our example, the p-sets for the verb phrases in (75.1) and (75.2) are essentially the sets of propositions that satisfy (76.1) and (76.2), respectively. The actual model-theoretic objects Rooth constructs are more complicated than this. The use of these values for $C$ in the translation rule (79) produces the following translations for the two verb phrases:

(80) 1. $\lambda x \forall P[[P(x) \;\&\; \exists y[P = {}^\wedge \mathit{introduce}'(y, \mathit{sue})]] \longrightarrow P = {}^\wedge \mathit{introduce}'(\mathit{bill}, \mathit{sue})]$

2. $\lambda x \forall P[[P(x) \;\&\; \exists y[P = {}^\wedge \mathit{introduce}'(\mathit{bill}, y)]] \longrightarrow P = {}^\wedge \mathit{introduce}'(\mathit{bill}, \mathit{sue})]$.

Rooth's theory accommodates multiple foci by generating an appropriate domain of quantification for phrases with multiple foci (Rooth, 1985, p. 61). For example, the p-set of the intension of the verb phrase

(81) [$_{VP}$ introduced Bill$_F$ to Sue$_F$]

is the set of properties of the form 'introduce $x$ to $y$'. This results in the correct meaning being assigned to (81): if John has a property of the form 'introduce $x$ to $y$', then it is the property 'introduce Bill to Sue.'

Rooth worries that his grammar doesn't handle correctly all cases of "embedded focus", that is, a focused phrase dominated by a node also having focus. For example, Rooth produces the same translation for both of the following sentences:

(82) 1. Jane ⟨sent a BOOK to MARY⟩.

2. Jane ⟨sent a/the book to MARY⟩.

The VP may optionally inherit focus from its NP$_F$ and PP$_F$ constituents in (82.1) and from its PP$_F$ constituent in (82.2). In either of these cases, the meaning of the sentence makes it an appropriate out-of-the-blue answer to a question about Jane's activities. In either case, the p-set of the VP is assigned the same fixed value: Rooth's definition of p-sets "does not pay attention to the p-sets of constituent phrases" of focused phrases (Rooth, 1985, p. 23). Because of this, the VPs in (82.1) and (82.2) receive the same translation; hence so do the two sentences.

In fact, Rooth cites Selkirk (1984) in explaining that (82.2) differs from (82.1) in that it is appropriate only in a context in which *a/the book* is "old information" (for instance, Jane's work involves books). Rooth appears to suggest that this shortcoming of his system's reflects a difference between the meaning of embedded and non-embedded focus. When focus is non-embedded, its meaning is explained in terms of the question the sentence answers and in terms of information from the non-focused part of the sentence. When focus is embedded, its meaning is explained in terms of novelty or givenness of discourse referents of the focused phrase:

> Embedded NP focus is linked to the novelty of the referent of the NP in the discourse ... The interpretation for an embedded focus has nothing to do with the role of the focused NP in the sentence as a whole; the opposite is true of non-embedded foci ... The role of the non-embedded focus in [ *John introduced* BILL *to Sue*] is to suggest that alternatives of the form 'John introduced *y* to Sue' are under consideration. The semantic object which implements this idea [a p-set] incorporates semantic information from the non-focused part of the sentence (Rooth, 1985, p. 24).

Rooth acknowledges that his proposal does not attempt to deal with this aspect of focus involving new/old discourse referents.

Rooth summarizes the advantages of his theory over the scope theory as the following (Rooth, 1985, p. 81):

- No syntactic bound variable is postulated; thus we do not expect the relation between *only/even* and the focused phrase to be restricted by scope islands or local conditions on variables.
- Association with multiple foci is accommodated without special stipulation.
- Association with focus is derived as a "theorem" from independently motivated principles.

A notable disadvantage of Rooth's theory is that, the way he exposes it, it accommodates the translation of *only* only in the construction [$_{VP}$ only VP]. This disadvantage is partly remedied by a modification he proposes in Chapter III of his dissertation that introduces a cross-categorial family of operators to handle *only* in several other constructions. Rooth's cross-categorial operators are described in Section 4.2 below.

## 2.6 NLU systems that handle focusing subjuncts

The only previous NLU system that treated focusing subjuncts was that of McCord (1986).[8] He calls them **emphatic** adverbs, members of the more general class of **focalizers**. He presents a semantic representation and semantic interpretation rules for focalizers, within the framework of a logic grammar developed earlier (McCord, 1982).

The class of focalizers includes (McCord, 1982, p. 224):

[8] The ideas in McCord's paper are partly reproduced in (Walker et al., 1987).

1. quantificational determiners like *all, some, most, many, few*, and *no.*

2. quantificational adverbs like *always, sometimes, usually, often, seldom*, and *never.*

3. the adverbs *only*, *just*, *also*, *too*, and *even.*

4. [non-lexical] discourse-functional modifiers dealing with topic/comment, yes/no question, contrast, etc.

In McCord's system, logical forms are used as the semantic representation. McCord proposes to represent the meaning of focalizers by a higher-order predicate with two arguments, the **base** and the **focus** of the focalizer. Each argument is in the form of an "open sentence", in fact a formula of first-order logic. For instance, the English sentences of (83) are assigned the logical forms shown in (84) by McCord's semantic interpreter.

(83) 1. Leopards often attack MONKEYS in trees.

2. All people like Mary.

3. Leopards only attack monkeys in TREES.

4. Is the King of France bald?

(84) 1. often(leopard($X$) & tree($Z$) & attack($E, X, Y$) & in($E, Z$), monkey($Y$)).

2. all(person($X$), like($E, X$,mary)).

3. only(leopard($X$) & monkey($Y$) & attack($E, X, Y$), tree($Z$) & in($E, Z$)).

4. yesno(def($X$, king($X$,france)), bald($X$)).

The meanings of the different focalizer predicates are distinct, but follow a rough pattern. For example, McCord explains the meaning of often(Base,Focus) as follows:

> Let B be the number of cases where Base is *true* and Focus is *defined* (has no presupposition failures), and assume B is finite. Let BF be the number of cases where Base&Focus is *true.* Then the meaning is that the ratio BF/B is "large." (McCord, 1986, p. 226)

The other quantificational adverbs, and the quantificational determiners, are analyzed similarly. For instance, *always* requires that BF/B = 1, and all(Base, Focus) means the same as always(Base, Focus).

McCord describes the sense of *only* as a focusing subjunct as follows:

> only(Base,Focus) presupposes that there is a case for which Base&Focus holds and asserts that for each case for which Base holds, Focus also holds (McCord, 1986, p. 228).

A *case* of only(Base,Focus) is defined as a tuple of values of the free variables appearing in the formula. This analysis is close to that of Horn (1969) and, says McCord, of Bergmann (1981). McCord's focus argument accords with the definition in Section 1.5 of the focus of a focusing subjunct. His base contains the same information as Karttunen and Peters's (1979) *scope* and Rooth's *p-set* of a sentence containing a focusing subjunct, and plays the same role in the definition of the meaning of *only*.

McCord notes the effect of stress in determining the focus of both the quantificational adverbs and the focusing subjuncts:

(85) 1. Leopards often attack monkeys in TREES.
2. often(leopard($X$) & monkey($Y$) & attack($E,X,Y$), tree($Z$) & in($E,Z$))

(86) 1. LEOPARDS often attack monkeys in trees.
2. often(monkey($Y$) & tree($Z$) & attack($E,X,Y$) & in($E,Z$), leopard($X$))

(87) 1. Leopards only attack monkeys in TREES.
2. only(leopard($X$) & monkey($Y$) & attack($E,X,Y$), tree($Z$) & in($E,Z$))

(88) 1. LEOPARDS even attack monkeys in trees.
2. even(monkey($Y$) & tree($Z$) & attack($E,X,Y$) & in($E,Z$), leopard($X$))

The input to McCord's semantic interpreter is a syntactic structure tree composed of complex *syntactic items* or *syns*. These are functors whose arguments include a list of (syntactic/morphological) *features*, a *predication*, an *operator*, and a list of *modifiers* (daughter *syns*). During the translation stage (see below) of semantic interpretation, each of the item's *modifiers* operates on the parent item's semantic form in sequence, beginning at the *end* of the list of modifiers. The *operator* determines how an item combines with its parent during translation to produce an intermediate semantic form. Semantic forms are composed of Operator–LogicalForm pairs, where the LogicalForm is constructed from the *predication* arguments of the *syns*.

Operators come in ten varieties. The abundance of operators allows a modifier to rearrange its modificand's logical form almost arbitrarily. The assortment of operators adds considerable manipulative power to McCord's semantic interpreter, but it also makes it more ad hoc. This added power may prove necessary for semantic interpretation, although McCord did not argue that this is so. His system lacks the intuitive uniformity of a Montague-style interpreter, which uses only one operator, functional application (McCord calls this the subst(X) operator). As well, there exists the possibility of an item's modifiers interfering with each other, as they operate serially to (sometimes drastically) rearrange their parent's logical form.

The semantic interpreter is broken into two stages. The first stage *reshapes* its input tree, in order to resolve the scope of each focalizer. McCord views the determination of focus as a scoping problem; he goes so far as to call the argument pair (Base,Focus) of a focalizer its *scope*. Scoping is taken care of entirely in this first stage. The second stage of the interpreter produces the logical form from the reshaped syntactic structure tree.

The mechanism to resolve focalizer scope is quite complex, and it depends heavily on a seemingly *ad hoc* assignment of *scoping types* and *precedence values* to each node in the tree. McCord provides no linguistic justification for the way in which scoping precedence values are assigned.

McCord's tree reshaping procedure results in the following algorithm for determining the focus of an focusing subjunct:

1. If the adverb's immediate right sister in the syntactic structure tree is a noun phrase, then the focus is that noun phrase. The resulting focalizer–focus pair is called a *focalizer group.*

2. Otherwise, any stressed right sister of the focusing subjunct is part of its focus, unless the sister is a focusing subjunct (other than one that is part of a focalizer group built by step 1).

3. Otherwise, the sister items of lower "precedence", together with any sisters that are focalizer groups built by steps 1 or 2, make up the focus. All items except other focalizers and noun phrases have lower precedence than a focusing subjunct.

A consequence of this algorithm is that the selection of an eligible focus from among a scale of constituents all containing a common stressed item is left to the syntactic component. Only the constituent that is a sister to the focusing subjunct can be its focus. Another consequence is the accommodation of multiple foci, as there is no requirement that only one constituent make up the focus of a focalizer.

## 2.7 Differences in coverage

The reviewed articles that specifically treat focusing subjuncts each cover a different set of constructions containing them. Their authors deal with different aspects of the syntax, semantics, and pragmatics of focusing subjuncts.

Horn deals with *even* and *only* modifying an NP. Using a different formulation, he also handles *only* modifying a VP, but only in case the focus of *only* is the entire VP. That is, he can handle (89.1) but not (89.2).

(89) 1. Muriel only ⟨voted for HUBERT⟩. (she didn't do the laundry)
2. Muriel only voted for ⟨HUBERT⟩. (she didn't vote for anyone else)

Horn purports to cover cases having just the verb as the focus of *only*, but this is shown in Section 2.2 not to be the case. He does not cover focusing subjuncts with multiple foci.

Jackendoff examines cases of *even* dominated by NP or by S (includes VP-modifying *even*). He deals with the phenomenon of multiple foci in a sentence, but his explanation of it doesn't cover all of the evidence. Jackendoff accepts the limit of one occurrence of *even* to a sentence.

Rooth's theory only covers *only* and *even* modifying a VP and having as focus any constituent within that VP. His theory distinguishes (89.2) from (89.1). It accommodates multiple foci of focusing subjuncts, though not all cases. In particular, embedded focus is not covered. Rooth's dissertation is a formal, model-theoretical account of the semantics of *even* and *only*. He leaves out questions of presupposition and assertion. When constructing a logical form for *only*, he variously omits the non-asserted meaning, or conjoins it with the asserted meaning.

Karttunen and Peters's analysis is limited to *even* in NP constructions. They further stipulate that the focused noun phrase must be interpreted *de re*. Their formal multi-part semantics distinguishes differences in the inheritance in complex sentences of *asserted* and *implicated* entailments of embedded phrases.

McCord's semantic interpreter apparently covers any occurrence of the focusing subjuncts *only*, *just*, *also*, *too*, and *even*. Multiple focusing subjuncts are not ruled out in a sentence. Multiple foci of focusing subjuncts are accommodated.

# Chapter 3

# Towards a semantics for focusing subjuncts

Before we can construct a semantics of focusing subjuncts, some of the issues outlined in Chapter 2 needed to be addressed; their resolution will be given in Section 3.1. The semantic framework that I employ and the syntactic behavior that the focusing subjuncts will be assumed to exhibit will be described in Section 3.2 and 3.3 respectively. The linguistic assumptions that my semantics will make will be summarized in Section 3.4.

## 3.1 Resolution of important issues

### 3.1.1 Meaning of *even* and *only*

The meaning of *even* and *only* set out by Horn (1969) constitutes a point of departure for research on focusing subjuncts. Recall that according to Horn's analysis,

(90) Only Muriel voted for Hubert.

means

(91)
1. Muriel voted for Hubert. [presupposed]
2. No one other than Muriel voted for Hubert. [asserted]

Horn's analysis assigns *only* a similar meaning when it modifies a verb phrase, the main difference being that one of the arguments to the *only* predicate is a property, making the predicate second-order.

Horn's description of *only* in its sense involving "expectation" is not followed up by any of the other literature. In this sense Horn takes *only* to be a three-place higher-order predicate whose arguments include a predicate (property) and a scale for ranking the strength of such properties.

Horn's core description of *even* and *only* is accepted in the literature, though often in a modified form. Rooth employs the notion of a "relevant property" in his description of the uniqueness assertion of *only* (when it modifies a VP). In Horn's version, *only* asserts that its first argument, the property $F$, is the only property true of some item $x$, whereas in Rooth's, *only* asserts that $F$ is the only *relevant* property true of $x$ (Rooth, 1985, p. 45). Much of Rooth's thesis describes how to obtain the set of these relevant properties. Fraser adds a third, pragmatic component in the case of the meaning of *even*: "The speaker would not expect or would not expect the hearer to expect [what is asserted]" (Fraser, 1971, p. 152). For example, the sentence

**(92)** Even MAX tried on the pants.

has a meaning composed of the three parts:

**(93)**
1. Max tried on the pants.
2. Other people tried on the pants.
3. The speaker would not expect or would not expect the hearer to expect Max to try on the pants.

Karttunen and Peters (1979) accept this description of *even*, interpreting the second and third components of the meaning in (93) to be conventional implicatures. Bergmann (1981, p. 121) treats these and Horn's "presupposition" of *only* as real presuppositions. However, she believes that (93.2) is additionally part of the *even* sentence's truth conditions (i.e. it is asserted). McCord takes *both* of the arguments of the focusing subjuncts to be properties ("open sentences"). The first argument is normally the property of being the item in the focus of the focusing subjunct, so the semantics amounts to the same as Horn's.

**Resolution** As explained in Section 2.2, Horn's formal rendering of the meaning of *only* is flawed. I will incline towards Rooth's and McCord's versions. According to Rooth, the effect that *only* has on a sentence that (without *only*) has logical form $a$ is

1. The sentence *conventionally implicates* $a$.
2. The sentence *asserts* that any "contextually relevant" proposition $P$ whose extension is true *is* the proposition $a$ (Rooth, 1985, p. 120).

The set of "contextually relevant" properties is the set of properties in the p-set (see Section 2.5) of $a$.

Rooth combines the conventional implicature and assertion of a sentence with *only* by conjoining them. I will keep these two components separate, so as to produce more informative output from my semantic interpreter for use, say, by a question-and-answer system (see below, Section 3.1.3).

Though there is controversy over the meaning of *even* and *only*, there is agreement that the meaning of a sentence containing these words is a function of two arguments: the focus and either a *scope* (Karttunen and Peters), *base* (McCord), or *p-set* (Rooth).

McCord defines the meaning of *only* by means of a higher-order *focalizer* predicate having two arguments, Base and Focus. A sentence represented using this focalizer

- presupposes that there is a case (a substitution for free variables) for which Base & Focus holds, and
- asserts that whenever Base holds, Focus holds too.

Now recall that the Focus formula is the logical form of the focus of *only*, normally a stressed constituent. Base is the remaining semantic material in the sentence. Recall also that the formula defining the p-set of a sentence is constructed analogously with its logical form, but with a variable substituted for the semantic content of the focus of only. So Rooth's p-set represents the same semantic content as McCord's Base argument. Similarly, Karttunen and Peters's *scope* argument contains the same information.

So the authors' semantics are equivalent. It matters not whether the meaning of *only* is described in terms of a sentence's logical form and its p-set, or in terms of a Focus and a Base or *scope*. The semantics introduced in this thesis will be modeled after Rooth's. I use his formulation in terms of p-sets.

The meaning I attribute to a sentence with *even* is, as per Rooth, that

- it asserts $a$, which is what the sentence would assert without *even*, and
- it presupposes that there is a true proposition $P$ in the p-set of $a$, which is not $a$ itself, and that $a$ is considered unlikely.

### 3.1.2 Pragmatic aspects of the meaning

The most obviously pragmatic aspect of focusing subjuncts is the expectational meaning introduced by *even* (inter alia). For simplicity, this component of the meaning of *even* is omitted in the analysis that follows in Section 4.2. Although Rooth describes it in terms of "likelihood", I don't feel that this fully captures the expectational component of *even*. Some reference to the speaker's and/or listener's knowledge and beliefs, which are contextual information falling within the purview of pragmatics, is required, as suggested by Fraser (1971, p. 153).

Rooth's formulation of the meaning of *even* and *only* in terms of "relevant properties" defined by p-sets suggests we consider these entities as a pragmatic aspect of meaning. Rooth's point of view is that the definition of p-sets is part of "the recursive component" of the linguistic theory and is independent of extra-sentential context; hence it is part of the semantics. However, he takes "the process which interprets the p-set associated with an S as a set of relevant propositions" as part of the pragmatic component of the theory (Rooth, 1985, p. 130).

It is my opinion that p-sets do contribute to the pragmatic processes that disambiguate *even* and *only* sentences that cannot be disambiguated except within a discourse context (see Section 3.1.6, below). On the other hand, there is no evidence that p-sets are properties or propositions that become discourse entities available in the context outside the sentence. For example, uttering

(94) John only eats ⟨RICE⟩.

which has the p-set

(95) John only eats something.

does not bring into being discourse entities concerning John's diet.

A final candidate pragmatic contribution of focusing subjuncts is their introduction of presuppositions and/or conventional implicatures, to which we now turn.

### 3.1.3 Presuppositions

The literature is in considerable conflict over the status of what Horn (1969) assumes are presuppositions (witness the title of his paper, *A presuppositional analysis of* only *and* even) introduced by *only* and *even*, namely (91.1) and (93.2) (similarly for the status of (93.3)). Fraser (1971, p. 153) argues that according to Austin's definitions (employed by Horn), these are instead "implications". That is, they are propositions whose falsehood does *not* preclude the sentence's having a truth value, as would be the case if they were presuppositions (per Austin's definition). Jackendoff accepts that they are presuppositions.

Karttunen and Peters propose that the non-asserted meaning introduced by *even* is a conventional implicature, a category that straddles the boundary between semantics and pragmatics. Their conclusion arises from their claim that the non-asserted meaning plays no role in determining the sentence's truth conditions. In order to deal in a general way with this kind of meaning, they propose a separate level of representation for conventional implicatures. They describe a sophisticated mechanism to permit them to be inherited by enclosing contexts (clauses) separately and differently from ordinary truth conditions (Karttunen and Peters, 1979, pp. 12, 16, 20). Bergmann (1981) also uses a multi-part semantic representation; however, she describes the second level of semantic conditions as representing presuppositions. Her theory is neutral on the question of whether or not they contribute to the sentence's truth conditions, as it is compatible with either position.

It should be noted that the conflict over the classification of non-asserted meaning reflects the lack of generally accepted definitions for terms such as "presupposition", "conventional implicature", and "pragmatics".

**Resolution** I remain uncommitted on the question of whether the non-asserted meanings of focusing subjuncts are presuppositions or conventional implicatures. Indeed, they may be both (or neither). The authors who favor labelling them as conventional implicatures, Karttunen and Peters (1979) and Rooth (1985), provide no indication that they are not also presuppositions. Rooth glosses over the difference between the two notions (Rooth, 1985, pp. 27–28).

In either case, the non-asserted meanings must be regarded as being pragmatic whether they are considered implicatures or presuppositions. Implicatures are by definition pragmatic inferences (see Section 1.5). Presuppositions involve reference to discourse participants' beliefs, as shown by Horton (1988).

No commitment is made to the exact status of non-asserted meaning. The approach taken in the current work is that asserted and non-asserted meanings are simply segregated. That is, a multi-part semantics is adopted along the lines of Karttunen and Peters (1979) and Bergmann (1981). This approach makes possible the correct handling of the inheritance of the non-asserted meaning of embedded sentences, which differs from the inheritance of asserted meaning. A major advantage is enhanced information provided to a consumer of the output of the semantic interpreter, such as a query-answer system. The distinction enables a question-answering system to differentiate between cases in which the answer is *no* because an asserted entailment is false, and cases in which it is *no* because a condition of the non-asserted meaning of the query is violated:

**(96)** Did Ross even ⟨wash the dog⟩?

1. No.
   [asserted meaning isn't true—Ross didn't wash the dog]
2. He did, but that's all he did.
   [non-asserted meaning is violated—he didn't do anything else]

The semantics described in this thesis assumes no pragmatic component involved in interpretation. The segregation of semantic representation makes the appropriate objects, those representing non-asserted meaning, available to be passed to any pragmatic component added in the future.

### 3.1.4 Distribution of the focusing subjuncts

None of the authors reviewed in Chapter 2 attempt to handle focusing subjuncts in their full distributional glory. Rather, focusing subjuncts are dealt with them in a limited number of the constructions in which they may appear. A few observations are made about the syntactic distribution of focusing subjuncts.

The writers are in partial accord regarding the allowable relative positions of a focusing subjunct and its focus. Anderson (1972, p. 899) notes that the focus of *even* must be a constituent adjacent to *even*, or be dominated by such a constituent. Jackendoff (1972, pp. 249–250) proposes that the focus be restricted to fall in a *range* that varies from one focusing subjunct to another. Jackendoff's definition of the range of *even* generates a restriction similar to Anderson's. The ranges of *only* and *just* are defined so that these words must precede their focus.

**Resolution** Previous research on focusing subjuncts deals with them only in constructions in which they modify an NP or a VP. However, as we saw in Section 1.1, focusing subjuncts can adjoin to any maximal projection, not just NP or VP. But there are restrictions that prevent adjunction of a focusing subjunct in a large number of cases. To begin with, it is uncontroversial that the focusing subjunct adjoins to a phrase that dominates its focus. I accept Jackendoff's observation that focusing subjuncts have individual ranges that specify where, relative to the focusing subjunct, its focus may be. For example, we saw in Section 2.3 that *only* has a range to its right and *alone* has a range to its left, whereas *even* has a range that includes nodes both to its right and to its left. I extend Jackendoff's observation to say that focusing subjuncts vary in their ability to *adjoin to* (not just find their focus in) the preceding or succeeding phrase.

There are constructions in which a focusing subjunct cannot normally appear. An example is in front of the NP object of a preposition:

**(97)** 1. * John has seen it near also [$_{NP}$ ⟨his back DOOR⟩].

2. * The library is closed on [$_{NP}$ only ⟨SUNDAY⟩].

But:

**(98)** 1. John hit the nail with only [$_{NP}$ ⟨a HAMMER⟩].

2. You can get a B grade for just [$_{NP}$ ⟨that answer⟩].
[Here *just* probably means *exactly* rather than *only*.]

The sentences (97) seem to be anomalous because the semantic role marked by the preposition is such that the prepositional object is unsuitable as focus of the focusing subjunct — the entire prepositional phrase, role (preposition) and filler (object) is semantically the suitable focus.

Rooth asserts that *even* and *only* are "bad" inside NPs:

**(99)** 1. *? The entrance only ⟨to the Santa Monica freeway⟩ was blocked off.

2. *? The entrance to only ⟨the Santa Monica freeway⟩ was blocked off.

I find the second (99) to be the more acceptable, even though *only* precedes the prepositional object.

Quirk observes that some focusing subjuncts have additional distributional restrictions with respect to their foci. For example, *just, merely, purely*, and *simply* must immediately precede their focus (Quirk et al., 1985, §8.118):

**(100)** * You can just get a B grade ⟨for that ANSWER⟩.

Sentence stress helps to determine sentence focus, and hence the focus of a focusing subjunct (see below, Section 3.1.5). Hence sentence stress also has an effect on the permissibility of a focusing subjunct at a given position.

For the purposes of this thesis, focusing subjuncts are assumed to be able to adjoin to any maximal projection. A given focusing subjunct is allowed to adjoin to the left or right, depending on its range (as per Jackendoff). This is certainly a simplification of the facts. The restriction on the range of a focusing subjunct can certainly involve more than just the notion of precedence. Jackendoff attributes a more complex range to *only*, in fact.

Other restrictions on the distribution of focusing subjuncts are disregarded, left for future consideration. My semantics will therefore be more flexible than actually necessary. An interpretation will be found for some ill-formed sentences, as well as for well-formed sentences containing focusing subjuncts.

### 3.1.5 The role of stress

All of the previous research acknowledges the effect of (intonational) stress in determining the meaning contributed to the sentence by *only* and *even*. Horn (1969, p. 100) alludes to the role of stress in disambiguating the focus of *only*. Fraser (1971, p. 165) notes that stress signals contrast in English. He attributes the presence of stress in the focus of *even* to the fact that part of the function of *even* is to contrast the focus with other possible similar items. These earlier authors assumed that the focus of *only* and the location of stress are directly related: that is, that the location of stress determines what the focus is. This relationship came to be seen as indirect, mediated via the focus feature (see Section 3.1.7, below). The later literature (Jackendoff, Rooth) talks more about the role of stress in signalling the presence of focus. Gussenhoven (1983) takes the signalling of the focus feature to be one of the functions of stress; he posits a further indirection in the link between focus and stress: stress signals the location of focus in a single "focus domain", which may contain *more* than one constituent with focus (Gussenhoven, 1983, p. 391). McCord, like the earlier writers, associates focusing subjuncts directly with stress. His semantic interpreter uses stress as the main factor in determining which constituent serves as the focus of a focusing subjunct (McCord, 1986, p. 238).

There is a dispute regarding the relationship between stress and the determination of the focus of a focusing subjunct that is orthogonal to the question of whether the relationship is direct or indirect. Anderson (1972) holds that stress falls at the end of the constituent serving as the focus of a focusing subjunct. Ladd (1983) argues that stress may fall before the end of the sentence focus (and hence of a focusing subjunct's focus). In McCord's semantic interpreter, several constituents may make up the focus of a focusing subjunct, and these constituents need not have final stress.

**Resolution** There is agreement that the location of stress helps determine the focus of *even* and *only*. For instance, stress partly disambiguates a sentence such as:

(101) John could also see his wife from the DOORway.

Readings such as

(102) * John could also see ⟨his wife⟩ from the DOORway.

are impossible, since the stressed constituent falls outside the candidate focus of the focusing subjunct. Jackendoff and Rooth take this observation a step further. Their proposals assume that stress *must* be present in the focus of the focusing subjunct (more precisely, that a linguistic marker of which stress is the overt realization must be there). This assumption, and Anderson's claim, which I accept, that stress always appears at the *end* of the focused item, reduces the available readings of a sentence to an ordered set in which each reading has a wider focus than the previous one:[1]

(103)
1. John could also see his wife from ⟨the DOORway⟩.
2. John could also see his wife ⟨from the DOORway⟩.
3. John could also ⟨see his wife from the DOORway⟩.

I will assume, like Jackendoff and Rooth, that the focus of a focusing subjunct obligatorily contains stress (but see Section 3.1.7, below).

### 3.1.6 Ambiguity

There is general agreement that the focus of a focusing subjunct is ambiguous, even though stress eliminates certain readings that would be possible in its absence (in written text, say). Specifically, the readings eliminated from consideration are those in which stress falls outside the focus (as in (104.1)) or before the end of the focus (as in (104.2)).

(104)
1. * John could also see ⟨his wife⟩ from the DOORway.
2. * John could also ⟨SEE his wife from the doorway⟩.

As was noted in Section 3.1.5, stress does not disambiguate completely.

We proceed on the assumption that there is a true ambiguity between readings of a sentence with different foci selected for the focusing subjuncts. That is, sentences such as (103) cannot be disambiguated by their overt form alone. This is predicted by the hypothesis that a focusing subjunct may adjoin to any constituent dominating its focus, not just to the focus itself. Hence any constituent of the phrase that a focusing subjunct adjoins to is eligible to be its focus (the sentences (11) in Section 1.1 illustrate this fact).

Sentences such as (101), and, in the absence of stress (such as in written text), (11), cannot be disambiguated except by interacting with the speaker, or perhaps by referring to the discourse context in which the sentences are uttered. For this reason, resolving this ambiguity is within the purview of the *pragmatics* of focusing subjuncts, not their *semantics*.

Note that p-sets can be helpful to the pragmatic process of disambiguating a sentence such as (103). For example, if the things that John could do (the p-set of (103.3)) are active

[1] *cf.* Wilson and Sperber (1979) who generate a set of ordered entailments by replacing variables for ever higher constituents of a sentence.

in the discourse, then reading (103.3) should obtain. If the places from which John could see his wife (the p-set of (103.1)) are in the discourse context, then reading (103.1) should obtain.

It is questionable whether there is any real ambiguity between the first two readings in (103). Clearly, (103.1) suggests that John could see his wife from other vantage points. Arguably, (103.2) does too. That is, contrast is suggested only with other possible fillers of the semantic role of *perspective*, not with fillers of arbitrary roles (as in *John could see his wife at 3 p.m.*). Therefore I take the position that no ambiguity obtains between a reading in which a PP is the focus and the reading in which just the object of the preposition is focused. Specifically, the PP will be taken to be the entity that can be focused. That is, the semantic material highlighted will be the semantic role indicated by the preposition plus its filler (a slot–filler pair in Absity nomenclature). In fact, it makes sense that when you draw attention to the object of the preposition, you are highlighting the fact that it fills whatever role the preposition indicates as well.

This treatment of PPs allows a consistent handling of other items whose semantic type is 'slot–filler pair'. In particular, adverbs have this type, but since they are single lexical items, they cannot be decomposed into a semantic role and a filler that is separately focusable.

### 3.1.7 Focus, the linguistic feature

The more recent writers, beginning with Jackendoff, assume the existence of a linguistic *focus* feature or marker. This binary feature may be marked at surface structure on any constituent. An earlier theory mentioned by Jackendoff (1972, p. 235) has focus marked at deep structure. Rooth adopts the assumption that focus is optionally "percolated" to syntactic phrases from their constituent phrases. That is, if a constituent has focus, then its parent may have focus as well. Focus mediates between stress and the focus of focusing subjuncts. That is, rather than being associated with stress, focusing subjuncts such as *even* are associated with focus, the feature, which must be present in the constituent(s) serving as the focus of the focusing subjunct (according to Jackendoff), or alternatively on its syntactic sister (argument) constituent (according to Rooth). Focus, in turn, is connected with stress, which is one of its overt realizations. Thus Fraser (1971) is really noticing the same thing as Jackendoff when he observes that stress is always present in the focus of *even*.

There is some discussion in the literature on whether focus has "meaning" of its own. Fraser attributes the link between stress and the focus of a focusing subjunct to the ability of stress to indicate *contrast*. This suggests that the focus feature has contrastive meaning. Jackendoff, on the other hand, attributes the compulsory presence of focus to the "unusualness" implication (e.g. (93.3)) introduced by *even* (1972, p. 165). Jackendoff identifies the focus feature with information status (new vs. given). Though Rooth does not dispute that focus is related to information status (Rooth, 1985, p. 10), his thesis treats focus as a purely syntactic feature whose presence has no obligatory semantic consequences. However, he acknowledges that information status of discourse referents is involved in the meaning of focus when focus is embedded (dominated by a node also having focus) (Rooth, 1985, p. 24). Gussenhoven (1983, p. 383) upholds the view that focus marks new information, specifically, "the speaker's declared contribution to the conversation." Dik et al. (1981, p. 42) take it to mark *salient* information. Ladd (1983, p. 166), on the other hand, cites examples of focus present on informationally "given" items.

**Resolution** Like Jackendoff's and Rooth's, my theory of the relationship between stress and focusing subjuncts talks about stress indirectly, mediated by mentioning a feature called *focus* (see Section 1.5) (this is Jackendoff's F marker, not his *focus*; see Section 2.3). Thus, strictly speaking, I will say that the focus of a focusing subjunct must contain the focus feature. Since its focus is always contained in the adjunct of a focusing subjunct, it follows that a focusing subjunct may only be adjoined to a phrase containing focus. Since the only realization of the focus feature that I accommodate is intonational stress, this is tantamount to stipulating that the focus of a focusing subjunct contains stress.

The theory could logically be extended to allow for other realizations of focus. For example, focus might be determined to be marked by default on the focusing node's adjunct constituent, in the absence of contrastive stress (such as in written text). This would be consistent with the evidence that in written text, the focusing subjunct is generally moved immediately adjacent to its focus rather than left next to a node that strictly dominates that focus (Quirk et al., 1985, §8.117, pp. 605–606):

**(105)** 1. John could see only ⟨his wife⟩ from the doorway.

2. John could see his wife only ⟨from the doorway⟩.

From this evidence, (106) may reasonably be determined to have the entire VP as its default focus.

**(106)** John could also [$_{VP}$ see his wife only from the doorway].

The notion of "the focus of a focusing subjunct" (see Section 1.5) will not actually be part of my formal semantics of focusing subjuncts. It should be noted, though, that the highest-level phrase that contains focus in the adjunct of the focusing subjunct will in fact correspond to that notion, i.e. the item the focusing subjunct draws attention to.

Like Rooth, I take the focus feature to be optionally and non-deterministically "inherited" by a node from its daughters. Rooth (1985, p. 21) proposes that focus may be acquired from the node's head or from daughters that serve as functional arguments. I observe that only those phrases that *end* with a stressed word are interpreted as being highlighted (Anderson, 1972, p. 899). This leads me to propose instead that focus is (optionally) acquired by a node from its *rightmost* daughter.

In my semantics, the extent to which focus non-deterministically percolates up the tree from the lexical item that is stressed affects the reading of a sentence by way of affecting the choice of the focus of the focusing subjunct. In Rooth's account, the nodes marked as focus are a given; the mechanism whereby a constituent acquires focus from a stressed lexical item it dominates is not addressed. Rooth did not note that the optional percolation of focus can explain ambiguity of a sentence with respect to the focus of its focusing subjunct.

I take no position on the question of whether or not focus is always marked on "new information". I attribute no semantic content to the focus feature itself, and assume no particular relationship between focus and information status.

### 3.1.8 Multiple foci of focusing subjuncts

The writers reviewed agree that a focusing subjunct can have within its focus two or more phrases (constituents) that do not together constitute a syntactic phrase (e.g., sentence 68 in Section 2.5). As Rooth points out, multiple foci present a problem for either of Anderson's or Fraser's theories. Rooth's theory is able to account for equal multiple foci. Jackendoff posits the existence of two subtypes of the focus feature, realized by two different pitch accents, in sentences with two foci (Jackendoff, 1972, p. 259).

**Resolution** My semantics accommodates focusing subjuncts that take two (or more) equal foci that are not, together, a single well-formed syntactic constituent. The mechanism is essentially the same as Rooth's (see Section 2.5, example (81)).

I take Rooth's concern to be misplaced regarding the difference between the sentences of (107) (see (82) of Section 2.5), which he says have varying "embedded" focus within a VP that is focused in both cases.

**(107)** 1. Jane ⟨sent a BOOK to MARY⟩.

2. Jane ⟨sent a/the book to MARY⟩.

In fact, I don't believe in embedded focus. With focus present as indicated in these sentences, they are both answers to questions about what different things Jane did. The speaker found it necessary to accent *book* in (107.1) but not in (107.2), even though (on the indicated reading) she didn't intend to contrast different sendings of objects, only different actions. These choices of accent convey pragmatic meaning independent of the semantics of focusing subjuncts. That is, as Rooth points out, the two sentences would be uttered in contexts differing with respect to how relevant or marked the fact is that Jane should send a book. I explain away the stress on *book* in (107.1), then, as not being the same focal-marking stress as that on *Mary*. In other words, focus is not really embedded; only the VPs in (107) are focused.

This argument has as a corollary that in sentences having several instances of stress that *does* mark focus, multiple-focus readings should occur. For instance, *even* in

(108) Jane even sent a BOOK to MARY.

should be interpreted as having the multiple foci *a book* and *to Mary*. The reading

(109) Jane even [$_{VP}$ ⟨sent a BOOK to MARY⟩].

should be prevented in cases where the stress on *book* is believed to mark focus. My theory will not accommodate this unless it forbids a constituent to acquire the focus feature if it has more than one focused daughter.

### 3.1.9 Multiple focusing subjuncts

The acceptability of sentences with multiple occurrences of *only* or *even*, including nested occurrences in which one falls within the other's scope, is controversial. Horn (1969) takes sentences such as (110) with nested *only* to be grammatical.

(110) Only John eats only rice.

He assumes that the presupposition and assertion of the inner *only* are inherited by the outer one in a straightforward way (see sentence (35) in Section 2.2). Fraser (1971, p. 168) accepts multiple occurrences of *even*, as in

(111) John can't sell even WHISKY to even INDIANS.

He believes semantic constraints make some sentences with multiple *even* ungrammatical. Anderson (1972, pp. 904–905), on the other hand, asserts that *even* is *always* limited to one occurrence per sentence, on semantic grounds that are "only hinted at" in his paper, but are related to the complexity of semantically interpreting such sentences. Jackendoff's theory restricts *even* to occur once per sentence. Rooth never mentions multiple occurrences of *only* or *even*.

**Resolution** I make the simplifying assumption that any one clause (S′) contains at most one focusing subjunct. This assumption is strongly challenged by evidence such as (110) and even more strongly by examples such as

(112) Even BILL eats only RICE.

and

(113) John persuaded even SUE that Bill only spoke FRENCH.

Usually, in cases such as these, the embedding is not within the "scope" (formally, the adjunct sister node) of the other focusing subjunct. The few examples I could come up with involved relative clauses:

(114) Even the man [$_S$ who only eats RICE] knows John.

Indeed, Rooth noted that focusing subjuncts are rarely allowed within NPs. In examples such as

**(115)** Even ⟨the house with only ⟨one⟩ bedroom⟩ was sold.

*only* performs a function that is not studied in this thesis: it combines with the determiner (*one*) to produce a new quantifier (*only one*).

### 3.1.10 Introduction of focusing subjuncts into sentence structures

There is a heated debate on how focusing subjuncts are introduced into sentence structures. Fraser (1971, p. 164) argues that *even* is introduced at deep structure, associated with the constituent serving as its focus. Then, movement in syntax can cause the two to be non-adjacent in surface form. Anderson (1972, p. 898) adopts the opposite position that *even* is freely introduced at deep structure, like any adverb, and becomes associated with its focus at a later derived stage, closer to surface structure. Jackendoff adopts the same position. Rooth criticizes both of the preceding theories for requiring movement of *even*, either in syntax or in the logical interpretation, that violates known constraints on movement. He advances yet a third theory, in which *only* and *even* do not take their focus as an argument at *any* stage of derivation. Instead, the focusing subjuncts take the adjacent constituent in S-structure as an argument, and it is this phrase that must have focus (Rooth, 1985, pp. 41–45). Rooth's arguments in favor of his theory are summarized in Section 2.5 of the present document.

Karttunen and Peters (1979) operate in a different theoretical framework, that of Montague Grammar. They introduce *even* into the sentence syncategorematically with a special *even rule* that combines the focus of *even* with a proposition (the scope of *even*, in the terminology of this thesis) to produce a transformed sentence. This results in *even*'s being associated with and adjacent to its focus in the single level of linguistic representation of their grammar. Of course, this means the grammar doesn't cover sentences in which *even* is not adjacent to its focus.

A related issue is whether the focus feature is assigned at deep structure, or at surface structure. Since it is generally agreed that *even* and *only* are associated with focused constituents, this issue subsumes the one described in the previous paragraphs. Jackendoff assumes that focus is a feature of surface structure. He rejects a *cleft theory* that takes sentences to be cleft at deep structure, the focus being the predicate of the higher clause (Jackendoff, 1972, pp. 230–231). Rooth only requires focus to be marked in S-structure (Rooth, 1985, p. 10).

**Resolution** Accepting Rooth's arguments, I take the syntactic argument of a focusing subjunct to be its adjunct (sister node), which I stipulate must contain focus. The focus of the focusing subjunct is not its argument, syntactically or semantically. Instead, information equivalent to knowing the focused item is obtained through Rooth's method of constructing p-sets. In my system, the p-set of a phrase is a semantic form, in the frame representation, constructed in parallel with the other semantic objects of the phrase.

Similarly, I take focus to be a feature marked at surface structure. In my opinion, the surface structure theory of the focus feature has more favorable consequences than the deep-structure theory for the automatic recognition of focus. If the former theory is true, focus is *defined* to be a phrase containing sentence stress in surface structure. If a deep-structure theory such as the one Jackendoff describes is adopted, then a deep structure for a sentence may be need to be constructed in order to prove that a phrase has focus.

## 3.2 Semantic framework

The avowed purpose of the present research is to develop a semantics for focusing subjuncts that is compositional and easily expressed in a frame-based meaning representation, such as that of Absity (described in Section 1.2.3). It is my specific goal to avoid the use of higher-order logics, such as Montague's IL (see Section 1.2.2), which are more expressive as representations of meaning but hence more intractable.

I shall in fact use Absity's semantic nomenclature to express the semantics of focusing subjuncts. Furthermore, the semantics of those parts of the grammar other than the focusing subjuncts will be essentially the same as Absity's. That is, the semantic framework in which the semantics of the focusing subjuncts is developed shares with Absity the properties described in Section 1.2.3.

The semantic framework differs in certain important respects from Absity's:

- Focusing subjuncts are not passive objects for composition, for reasons introduced in Section 1.3.
- Typing is not as strong. *Even* is allowed to combine during parsing with phrases having various different semantic types. This is because *even* can adjoin to phrases represented in the Absity scheme as slot–filler pairs as well as to other phrases represented as frame descriptions. Sentences can be represented by one of *two* possible types, the frame statement or the *frame implication* (to be described in Section 4.2), a new type introduced into the semantic nomenclature in order to handle focusing subjuncts.
- A multi-part semantic representation is used. Several objects, rather than just one, represent the semantic content of a phrase (these will be elaborated on in Section 4.2):
  - the assertional contribution of a phrase,
  - its non-asserted or "presuppositional" contribution,
  - its p-set (Rooth's notion, see Section 2.5), and
  - a latent focusing subjunct operator.

## 3.3 Syntactic framework

The semantics described in this thesis does not depend on the adoption of any single choice of a syntactic framework or theory. It is my intention to provide a semantics compatible with any syntactic theory suited to compositional semantics. I will express the syntax of focusing subjuncts in the formalism of Generalized Phrase Structure Grammar (GPSG) (Gazdar et al., 1985), but they could equally well be described relative to another view of syntax, such as the theory of Government and Binding (GB) (Chomsky, 1981). Nothing in the semantics of focusing subjuncts being propounded depends on a particular syntactic theory, but GPSG is the syntactic framework of choice for several reasons:

- GPSG uses a compositional semantics (which Sells (1985, p. 97) calls *extended Montague Grammar*).

- GPSG grammars resemble Definite Clause Grammars, which are said to be easy to understand and to implement. Evans (1985) and Phillips and Thompson (1985) show how the rules of a grammar expressed in the GPSG formalism can be converted into a set of context-free rules.

- A central notion of GPSG is that of the *features* of a syntactic phrase.

The last of these, having the features of a phrase explicitly accessible, is useful for several reasons. Firstly, as was seen in the literature survey (Chapter 2), a crucial input to the semantic interpretation of focusing subjuncts is the binary focus feature that may be marked on a syntactic node. Secondly, it will be convenient to associate semantic objects with a phrase by making them values of appropriate features on the *category* for the phrase. This is an extension of the approach taken by Lexical-Functional Grammar, a theory in which the semantics of a phrase is carried on its functional structure (similar to a GPSG category) as the value of its *pred* attribute (Sells, 1985, p. 145). The same approach is taken by Shieber's (1986) PATR-II formalism for unification-based grammars. Shieber asserts that GPSG is a unification-based formalism amenable to being expressed in the PATR-II formalism.

Readers unfamiliar with the GPSG theory are referred to Sells (1985) for an introduction, and to Gazdar et al. (1985) for the full version of the theory. The essential facets of the theory relevant to this thesis are this following:

- Information about syntactic nodes is encoded by means of syntactic features. A feature is a piece of linguistic information, such as *tense, number*, and *bar level.* A feature may be *atom-valued* or *category-valued.*

- A category is a set of feature–value pairs, that is, a feature structure. The notation for a feature–value pair is ⟨FEATURE, value⟩ in GPSG. For example, *NP* in GPSG is an abbreviation for the category {⟨N,+⟩, ⟨V,−⟩, ⟨BAR,2⟩}; any category having this set of feature–value pairs as a subset is an NP (Sells, 1985, pp. 80–83).

- A grammar includes a set of *Immediate Dominance* (ID) rules specifying what categories can dominate other categories in a parse tree. *Metarules* are used to construct additional ID rules from the basic set.

- Feature Co-occurrence Restrictions (FCRs) constrain the allowable categories for a language. For example, the FCR that Gazdar et al. (1985) call FCR1 imposes the constraint that if a category is marked as being inverted, then it must also be marked as an auxiliary and tensed.

  (116) *FCR 1:*
  $[+\text{INV}] \supset [+\text{AUX},\text{FIN}]$

  According to Sells (1985, p. 103), "this expresses the fact that in English, the only things that invert are (tensed, or finite) auxiliary verbs".

- Feature Specification Defaults (FSDs) specify "unmarked" values for features. FSD1 says that things are not inverted unless specifically forced to be, for instance by an ID rule.

  (117) *FSD 1:*
  $[-\text{INV}]$

- Three principles, the Head Feature Convention (HFC), the Foot Feature Principle (FFP), and the Control Agreement Principle (CAP), regulate the flow of information (contained on features) throughout the syntax tree, thus acting as well-formedness conditions on trees (Sells, 1985, p. 79).

- Linear Precedence (LP) Statements are conditions on trees that specify the order in which sibling nodes may appear.

The HFC states, essentially, that the **head** features of the mother category in an ID rule (equivalently, in a local syntax tree) are identical to the head features of the head daughter. Head features include N, V, BAR, SUBCAT, AGR, and many others. ID rules and FCRs often dictate that features like BAR and SUBCAT be different on the mother and head daughter; the HFC allows for this.

The FFP states, roughly, that **foot** features are passed up from *any* daughter in a tree, not just the head. The CAP dictates that functor categories—such as a VP—agree with argument categories, their **controllers**—for instance, the subject NP. Agreement is accomplished by means of the AGR feature which, on the controllee, carries a "copy" of the controller. For example, a VP that has the agreement feature ⟨AGR, NP[PER 3, +PLU]⟩ will *agree* with a subject NP category that includes the feature–value pairs {⟨PER,3⟩, ⟨PLU,+⟩}.

In the theory of GPSG, FCRs and FSDs are considered to be language-specific, though some FCRs are presumed universal. The HFC, FFP, and CAP, and LP Statements are

| Short form | Category |
|---|---|
| S′ | {⟨N,−⟩, ⟨V,+⟩, ⟨BAR,2⟩, ⟨SUBJ,+⟩, ⟨COMP,α⟩}<br>where α ∈ {*that*, *for*, *whether*, *if*} |
| S | {⟨N,−⟩, ⟨V,+⟩, ⟨BAR,2⟩, ⟨SUBJ,+⟩, ⟨COMP,NIL⟩} |
| XP | {⟨BAR,2⟩} |
| VP | {⟨N,−⟩, ⟨V,+⟩, ⟨BAR,2⟩, ⟨SUBJ,−⟩, ⟨COMP,NIL⟩} |
| VP[BSE] | {⟨N,−⟩, ⟨V,+⟩, ⟨BAR,2⟩, ⟨SUBJ,−⟩, ⟨COMP,NIL⟩, ⟨VFORM,BSE⟩} |
| NP | {⟨N,+⟩, ⟨V,−⟩, ⟨BAR,2⟩} |
| PP[α] | {⟨N,−⟩, ⟨V,−⟩, ⟨BAR,2⟩, ⟨PFORM,α⟩} |
| H[*n*] | {⟨SUBCAT,*n*⟩}<br>A category abbreviated as "H" is a head daughter. |

Figure 3.1: Some GPSG category abbreviations

considered to be principles of Universal Grammar "that may be subject to finite variation across languages" (Sells, 1985, p. 101).

ID rules in GPSG can appear cryptic because they leave out information that is implied by the FCRs, LP Statements, and the principles, especially the HFC. As well, GPSG uses abbreviations for certain common sets of features. Some GPSG abbreviations that appear in Chapter 4 of this thesis are explained in Figure 3.1. Consider for example the ID rule that says that a sentence node consists of a subject followed by a verb phrase:

(118) S ⟶ XP, H[−SUBJ]

H[−SUBJ] stands for a category that is marked ⟨SUBJ,−⟩ and is the head of the sentence. Therefore, because of the HFC, this category shares with S the head feature–value pairs ⟨N,−⟩, ⟨V,+⟩, ⟨BAR,2⟩, and ⟨COMP,NIL⟩. This makes H[−SUBJ] a verb phrase, since "verb phrase" (abbreviated "VP") is the category consisting of these four feature–value pairs plus the pair ⟨SUBJ,−⟩. Given this information, and knowing the abbreviations "S", "XP", and "H", the ID rule (118) states that an S-node can consist of any ⟨BAR,2⟩ phrase (i.e., any maximal projection; this allows for non-NP subjects) and a VP. Note that the ID rule does not specify the order in which the subject and VP appear in the sentence. This is the responsibility of the LP Statements.

Rule (119) is an example of a **lexical** ID rule.

(119) VP ⟶ H[5], NP, NP

This type of ID rule introduces a **lexical head**, indicated by the notation "H[*n*]", which is an abbreviation for {⟨SUBCAT,*n*⟩}. There is an FCR that forces such categories also to have the BAR feature defined. These rules "provide the arguments subcategorized for by heads" (Sells, 1985, p. 89). The rule (119) says that a VP consists of a verb with SUBCAT value 5 (the HFC requiring that only a verb head a VP), and two noun phrases (NPs).

Bi-transitive verbs that are allowed in the structure of the ID rule (119) are assigned a category in the lexicon that includes the feature–value pair ⟨SUBCAT,5⟩. The SUBCAT feature is GPSG's particular way of implementing subcategorization: it is an indicator on each verb, noun, preposition, and adjective in the lexicon that points to the structures in which they may occur (Sells, 1985, p. 87).

## 3.4 Summary of assumptions

### 3.4.1 Choices of formal frameworks

- Higher-order logics such as Montague's IL are intractable. A frame formalism such as that used by Absity is less expressive, but more tractable.

- It is acknowledged that second-order constructs are necessary to express the meaning of certain linguistic constructs. However, I show in this thesis that $\lambda$-expressions, in particular, are unnecessary to represent the semantics of focusing subjuncts.

- The semantics described in this thesis operates with the GPSG theory of grammar. However, I do not believe that the semantics is inconsistent with other theories, GB theory among them.

### 3.4.2 Simplifying assumptions

The following list of the simplifying assumptions that I make with respect to the syntax and semantics of focusing subjuncts was promised in Section 1.4.1.

- The semantics I describe assumes that intensions need not be represented; they are omitted to gain efficiency and notational simplicity. Therefore, it does not handle intensional ambiguity. Note that though Rooth (1985) *does* represent intensions in his semantic translation, he uses them only to obtain the notion of *properties* (expressed for instance by a verb phrase). My semantics does not require an explicit representation of these properties (see Section 4.2).

- Although it is considered an important component of the meaning of *even*, the expectational meaning it introduces is not explicitly part of the semantics detailed in Section 4.2.

- The semantics and the implementation outlined in Chapter 5 do not have a pragmatic component that takes into account context and discourse focal entities. It is assumed, however, that such a component would take advantage of the separate semantic objects I obtain for the asserted and non-asserted meanings of sentences containing focusing subjuncts.

- Only a few syntactic restrictions are imposed on focusing subjuncts. This simplifies their syntactic treatment. They are required to occur in adjunction structures, adjoined to a maximal projection. The focus of the focusing subjunct must be contained within this phrase, within some *range* (see Section 3.1.4) that varies somewhat from one focusing subjunct to another. The semantics described herein is intended to be general enough to account for focusing subjuncts in any such structure. Certainly the syntax of focusing subjuncts is far more complex than this. However, it is difficult to systematically determine when a focusing subjunct may not adjoin to a maximal projection. Certainly, Quirk et al. (1985) do not do so. A full account of the syntax of focusing subjuncts is a subject for further research.

- I do not deal with focusing subjuncts that modify determiners to produce quantifiers such as *only one*.

- Like Rooth (1985) and Jackendoff (1972), I assume that the focus of a focusing subjunct (the part of the sentence to which attention is drawn) is obligatorily stressed. Now in fact, my theory really licenses no more of an assumption than that the focus should be marked with the linguistic focus feature. Realizations of this feature other than stress exist, but they are not dealt with here.

- Clauses containing more than one focusing subjunct are not dealt with. There is evidence that such sentences are well formed.

- The treatment of intonation is simplified to the bare bones. I assume that there is only one intonational parameter: a word is either stressed, or it is not. In fact, detecting intonational stress in speech is a non-trivial matter, and there is a much richer structure to intonation than is suggested by my assumption. The reader is referred to Hirschberg and Pierrehumbert (1986), Pierrehumbert (1980), and Jackendoff (1972, p. 259).

- The focusing subjuncts are assumed to have meanings based on those of *even* and *only*. Understanding the semantics of these two key words takes us most of the way towards understanding all of the focusing subjuncts.

### 3.4.3 Linguistic points

- P-sets are *not* assumed to correspond to the set of discourse entities introduced by a sentence. Because of this and the fact that they are computed compositionally from just the form of the sentence, they are not taken to be a pragmatic notion.

- No commitment is made on whether the non-asserted meanings of focusing subjuncts are in fact presuppositions.

- The same reading for a prepositional phrase arises when the whole phrase is [+ focus] as when just the object of the preposition is. No ambiguity of meaning is detected as a result of such a contrast. I make the choice that in either of these cases, the semantic content of the whole prepositional phrase (a role and its filler, in Absity terms) is the focus of the focusing subjunct.

- Following Rooth (1985), I take the focus feature to be optionally acquired by a node from its daughter. The observation that stress always occurs at the end of a focused constituent leads me to propose different conditions for the "percolation" of focus than Rooth does: focus may be inherited by a node from its rightmost daughter. Within the syntactic framework of GPSG, focus appears as a head feature, but some manipulation of the ID rules of a grammar are required to obtain the desired behavior of the feature.

- No position is taken on the meaning of the focus feature itself. I do not require that it mark only information "new" to a discourse.

- A focusing subjunct may take multiple foci that do not together constitute a well-formed syntactic phrase.

# Chapter 4

# A compositional semantics for focusing subjuncts

In Section 4.2, a compositional semantics for the focusing subjuncts *even*, *only*, and *too* will be defined. First, the syntax with which this semantics operates in tandem is defined in Section 4.1.

The following typographical conventions are employed in this chapter and Chapter 5: a sans-serif typeface is used for technical terms that are part of my theory (e.g. focus-in). GPSG features appear in capital letters (e.g. BAR). The GPSG notation is used for feature–value pairs (e.g. ⟨focus, +⟩).

## 4.1 The syntax of focusing subjuncts

As mentioned in Section 3.3, my starting point is a GPSG grammar that handles some simple VP, NP, adjective and adverb constructions in the manner described by Gazdar et al. (1985). I introduce the focusing subjuncts by means of a rule that allows a focusing subjunct to adjoin to any maximal projection, subject to the restrictions shown in (120). Note that in GPSG, the maximal projections, those phrases with bar level 2, are S′, S, VP, NP, AP, and PP (Sells, 1985, p. 82).

**(120)** *Conditions on focusing subjunct adjunction:*

1. The adjunct must contain a focused constituent.
2. The focusing subjunct may precede the adjunct if its *range* (Jackendoff-style) includes its right sister, or follow the adjunct if its range includes its left sister. Range varies from one focusing subjunct to another.

The second condition follows from the first one, together with Jackendoff's *Association with Focus* rule, which stipulates that the focus of a focusing subjunct, which must be [+ focus], lies within the range of the focusing subjunct.

It is conceded that my conditions on the introduction of focusing subjuncts are insufficient and will over-generate, sanctioning sentences such as (97), (99) and (100) (Section 3.1.4). Furthermore, the simple specification of the range doesn't permit complex focusing subjunct ranges proposed by Jackendoff (1972, p. 251):

**(121)** *Range of* even:
If *even* is immediately dominated by a node X, the range of *even* includes X and all nodes dominated by X to the right of *even*, plus the subjunct if X is an S.

As long as my semantics can cope with an overgenerated set of syntactic constructions, it will certainly be able to deal with the subset of these constructions that is determined to be well-formed by a more precise syntax.

My syntax cannot recognize structures in which the focusing subjunct is a sentence adverb dominated by an S node. Rooth allows *even* to occur in this construction (Rooth, 1985, pp. 99, 122):

**(122)** [$_S$ [$_{NP}$ ⟨JOHN⟩] even [$_{VP}$ came]].

I interpret the sentences to which he assigns this structure as cases of subject-NP or VP adjunction.

Five new features are added to GPSG and are employed to realize the introduction of focusing subjuncts outlined above: focus, focus-in, range-left, range-right, and fs.

### 4.1.1 Focus in GPSG

Focus is a binary feature indicating whether or not linguistic focus (see Sections 1.5 and 3.1.7) is marked on a category. I wish to make focus a head feature, that is, a feature that is percolated from a head daughter to its parent node. Focus originates on stressed input words. A morphological component adds ⟨focus +⟩ to a category obtained from the lexicon for a word that is stressed.

Now, unfortunately, the flow of GPSG head features in a syntax tree does not accord with the behavior that Rooth attributes to the focus feature (see Sections 3.1.7 and 2.5). Nor does it accord with the observation that stress comes at the *end* of a focused constituent, so that focus should be percolated only from a rightmost daughter. In effect, I require focus to be a special kind of head feature. It can percolate to a parent not just from the head daughter, but also from a daughter that serves as "functional argument" to the daughter. In the latter cases, it appears, focus may optionally fail to percolate to its parent node. In addition, I require that it be able to percolate to a parent node only from its rightmost daughter. Defining exactly this new type of head feature requires a modification to the GPSG theory, notably to the Head Feature Convention, which the reader may be reluctant to accept. A similar effect can however be accomplished within the existing GPSG theory, albeit in a somewhat clumsy manner, by using the ID rules as follows.

Let focus be a head feature, so that it percolates to its parent node from its head in each ID rule that does not specifically mention the feature. For those ID rules in which some daughter node serves as a functional argument—and occurs rightmost in the surface form (this requires reference to the LP statements)—allow focus to be percolated by making a near-duplicate ID rule in which focus is instantiated on the daughter in question and on the parent. If the ID rule constitutes a point in the grammar at which the difference between a daughter having or lacking focus can create a real ambiguity (see Section 3.1.6), add a further ID rule in which the daughter is specified as ⟨focus +⟩, but not the parent. By the mechanism of the HFC, focus may or may not be percolated to the parent node, depending on whether the old or new ID rule is used.

This mechanism allows for a difference in the percolation of focus at those places at which an ambiguity may arise with respect to the focus of a focusing subjunct. For instance, we are indifferent as to whether or not focus percolates to a PP from its prepositional object—the same reading obtains in either case. Therefore, the ID rule

**(123)** PP ⟶ P, NP

will be paired only with the first kind of near-duplicate ID rule, not the second kind, by the modification of the grammar proposed above:

**(124)** PP[+ focus] ⟶ P, NP[+ focus].

Hence focus will always percolate to PP if it is marked on NP. On the other hand, there is a difference between a reading in which a VP is focused and the reading in which only its direct object is focused. So the ID rule (125.1) expands to the three ID rules in (125).

**(125)**
1. VP ⟶ V, NP
2. VP[+ focus] ⟶ V, NP[+ focus].
3. VP ⟶ V, NP[+ focus].

### 4.1.2 The other new features

**Focus-in** is a binary foot feature introduced onto a node wherever focus is, by means of a hypothesized Feature Co-occurrence Restriction. Any node dominating a stressed lexical item will be ⟨focus-in +⟩. This is the requirement for a node to be the adjunct of a focusing subjunct (see Section 3.1.5).

**Range-left** and **Range-right** are non-propagating binary features marked on the lexical entries for focusing subjuncts. A focusing subjunct that is ⟨range-right +⟩ may adjoin in front of its adjunct, and one that is ⟨range-left +⟩ may adjoin following it. This is accomplished by means of Linear Precedence rules that assign maximal precedence to the former kind of focusing subjunct and minimal precedence to the latter. A focusing subjunct that is marked with *both* of these features is not constrained at all by the LP rules: it may occur before or after its single sister node, its adjunct.

**Fs** is a foot feature that records the semantic value of the focusing subjunct that a node dominates, or the value "–" if none. It is introduced by being present on focusing subjunct lexical entries.

There are three additional new features used and manipulated by the semantics: the **presupp**, **assert**, and **p-set** features, which will be described in Section 4.2.

### 4.1.3 Illustrative grammar

The following Feature Co-occurrence Restrictions and Feature Specification Defaults apply to the new features:

(126) *Feature Co-occurrence Restrictions:*

1. $\neg$ ($\langle$fs $\alpha\rangle$ & $\langle$range-left $-\rangle$ & $\langle$range-right $-\rangle$)
2. $\langle$focus $+\rangle \supset \langle$focus-in $+\rangle$

The FCR (126.1) just says that a focusing subjunct has range either to its left or to its right (or both). (126.2) says that the feature **focus-in** is marked whenever **focus** is marked.

(127) *Feature Specification Defaults:*

1. [– focus]
2. [– fs]

That is, the default values of the **focus** and **fs** features is that they are not marked.

The following toy grammar will be used for illustrative purposes.

**Metarules:** The Subject-Aux Inversion (SAI) Metarule (Gazdar et al., 1985, p. 62).

**ID rules:** Figure 4.1 shows the ID rules that incorporate the effect of the SAI Metarule. The rules are numbered in order that we may easily refer to them, particularly when defining their corresponding semantic rules. The numbering scheme is idiosyncratic; no significance is to be attached to the number assigned to a particular ID rule. For a brief introduction to the notation for expressing ID rules in GPSG, see Section 3.3.

**LP Statements**

1. $\{\langle$fs $\alpha\rangle$, $\langle$range-right $+\rangle$, $\langle$range-left $-\rangle\}$ $\prec$ X
2. X $\prec$ $\{\langle$fs $\alpha\rangle$, $\langle$range-right $-\rangle$, $\langle$range-left $+\rangle\}$

These Linear Precedence statements cause a focusing subjunct marked **range-right** to precede its sibling, and one that is marked **range-right** to follow its sibling.

**Lexicon:** Figure 4.2 gives lexical entries for the focusing subjuncts *even*, *only*, and *too*, and for selected other words.

| Rule number | Rule | Comment |
|---|---|---|
| 101 | S ⟶ XP, H[−SUBJ] | H[−SUBJ] is a VP |
| 1 | VP ⟶ H[1] | |
| 2 | VP ⟶ H[2], NP | |
| 3 | VP ⟶ H[3], NP, PP[to] | |
| 5 | VP ⟶ H[5], NP, NP | |
| 131 | NP ⟶ H[+R] | |
| 130 | NP ⟶ H[130] | H[130] is a proper noun |
| 30a | NP ⟶ Det, $H^1$ | |
| 132 | NP ⟶ Det, $H^1$, S[+R] | S[+R] is a relative clause |
| 30 | $N^1$ ⟶ H[30] | |
| 38 | PP ⟶ H[38], NP | |
| 107 | VP[+AUX] ⟶ H[107], VP[BSE] | H[107] is an auxiliary |
| 207 | VP[+INV, +SUBJ, +AUX] ⟶ NP, H[107], VP[BSE] | produced from 107 by SAI Metarule |

Figure 4.1: ID rules for toy grammar

| Lexical item | Category |
|---|---|
| even | {⟨fs even⟩, ⟨range-left +⟩, ⟨range-right +⟩, ⟨BAR −⟩} |
| only | {⟨fs only⟩, ⟨range-left −⟩, ⟨range-right +⟩, ⟨BAR −⟩} |
| too | {⟨fs too⟩, ⟨range-left +⟩, ⟨range-right −⟩, ⟨BAR −⟩} |
| sleeps | {⟨N −⟩, ⟨V +⟩, ⟨BAR 0⟩, ⟨SUBCAT 1⟩, ⟨AGR {⟨NUM SING⟩, ⟨PERS 3⟩}⟩, ⟨assert (sleep ?*x*)⟩} |
| wash | {⟨N −⟩, ⟨V +⟩, ⟨BAR 0⟩, ⟨SUBCAT 2⟩, ⟨AGR {⟨NUM PL⟩, ⟨PERS 3⟩}⟩, ⟨assert (wash ?*x*)⟩} |
| does | {⟨N −⟩, ⟨V +⟩, ⟨BAR 0⟩, ⟨SUBCAT 107⟩, ⟨VFORM AUX⟩, ⟨AGR {⟨NUM SING⟩, ⟨PERS 3⟩}⟩, ⟨assert (tense continuous)⟩} |
| to | {⟨N −⟩, ⟨V −⟩, ⟨BAR 0⟩, ⟨SUBCAT 38⟩, ⟨PFORM to⟩, ⟨assert (recipient ?*y*)⟩} |
| the | {⟨BAR −⟩, ⟨AGR {⟨NUM PL⟩, ⟨PERS 3⟩}⟩, ⟨assert (the ?*x*)⟩} |
| men | {⟨N +⟩, ⟨V −⟩, ⟨BAR 0⟩, ⟨SUBCAT 30⟩, ⟨AGR {⟨NUM PL⟩, ⟨PERS 3⟩}⟩, ⟨assert (person ?*x* (gender masc))⟩} |
| dog | {⟨N +⟩, ⟨V −⟩, ⟨BAR 0⟩, ⟨SUBCAT 30⟩, ⟨AGR {⟨NUM SING⟩, ⟨PERS 3⟩}⟩, ⟨assert (dog ?*x*)⟩} |

Figure 4.2: Lexical entries for toy grammar

### 4.1.4 Introduction of focusing subjuncts

Focusing subjuncts are introduced by the following ID rule:

**(128)** *Focusing subjunct ID rule:*

111 XP[fs $\alpha$, +focus-in] $\longrightarrow$ $\{\langle \text{fs } \alpha\rangle, \langle \text{bar } -\rangle\}$, H[+focus-in]
where $\alpha$ is one of *even*, *only*, *too*.

The first of the two categories mentioned on the right-hand side of this rule can only be obtained from a lexical entry for a focusing subjunct.

The lexical entries for the focusing subjuncts are relatively impoverished, having only the **range-left**, **range-right**, and **fs** features defined on them. They lack semantic features (except for **fs**) and the common syntactic features N, V, and SUBCAT (they're not heads). They are effectively introduced syncategorematically, having no semantic type, except for the sentential operator implicit in the **fs** feature (see Section 4.2), and no syntactic type except insofar as only lexical items that are $\langle \text{fs } \alpha\rangle$ are picked up by the lexical ID rules that introduce focusing subjuncts.

### 4.1.5 Pseudo-prepositions

A final adjustment is made to the syntax for the treatment of subjects and of NPs subcategorized by VPs. I want the subject, object, and indirect object of a verb to be functional arguments of the verbs (or slot–filler pairs to the frame representing the verb, in Absity terminology).

In the toy grammar just given, the object and indirect object are NPs and the subject may be an NP. Recall that NPs have the semantic type of a frame description, not of a slot–filler pair (see Figure 1.1). I will take Absity's approach to overcoming this problem. Syntactic processing will be assumed to insert the case flags **subj**, **obj**, and **iobj** in front of the appropriate NPs, for the benefit of the semantics. The flags will be treated as prepositions; I call them **pseudo-prepositions** (per Hirst (1987)). Indeed, the case-marking effect of the syntactic positions of subject, etc., is similar to the role of a preposition. It is quite natural to treat the semantic contribution of syntactic position as equivalent to that of a preposition. As in Absity, subjects and objects are seen by the semantics as PPs, which have the desired type, that of slot–filler pairs (Hirst, 1987, pp. 50–51).

I accomplish the insertion of pseudo-prepositions by modifying the ID rules for S and VP categories as shown in Figure 4.3, and by inserting the appropriate pseudo-preposition in front of the word list to be consumed by each new PP. Note that in replacing the rule numbered 101, I have lost the power to generate subjects that are not NPs, such as in this example (Gazdar et al., 1985, p. 113):

**(129)** $[_{S'}$ For us to be late] would bother Lee.

| Rule number | Rule | Comment |
|---|---|---|
| 101 | S ⟶ PP[subj], H[−SUBJ] | |
| 2 | VP ⟶ H[2], PP[obj] | |
| 3 | VP ⟶ H[3], PP[obj], PP[to] | |
| 5 | VP ⟶ H[5], PP[iobj], PP[obj] | |
| 207 | VP[+INV, +SUBJ, +AUX]<br>⟶ PP[subj], H[107], VP[BSE] | produced from 107 by SAI Metarule |

Figure 4.3: Revised ID rules

## 4.2 The semantics of focusing subjuncts

In this section, I review Rooth's semantics for *even* and *only* (see Section 2.5) and propose an adaptation to it within an Absity-like frame-based compositional semantics framework (see Sections 1.2 and 3.2). The semantics is described in relation to the syntax described in Section 4.1.

### 4.2.1 Background

A main feature of the proposed semantics is that it arrives at the final translation of a sentence containing *even* or *only* without employing $\lambda$-expressions in any of the intermediate steps. I show how $\lambda$-expressions may be dispensed with without sacrificing expressive power. This is desirable to avoid having to define constructions for $\lambda$-expressions in a frame representation, which are troublesome and make the frame language less tractable. As will be explained, this is accomplished by recognizing that the semantics of focusing subjuncts actually takes place at the sentence level. That is, the semantic representation of a focusing subjunct is a *sentential* operator, one that produces a new semantic object from the semantic representation of a *sentence*.

Note that although $\lambda$-expressions are dispensed with, my semantics still employs a semantic representation of *even* and *only* that requires an extension to Absity's frame representation system. The semantic representation of a focusing subjunct is a function or operator on a semantic object, rather than one of the usual Absity types—slots, frames, etc.—that serve as passive objects for composition.

In order to show how it is possible to represent the semantics of focusing subjuncts without $\lambda$-expressions, we examine Rooth's semantics of *only* and *even*. Recall that Rooth's translation rule for *only* in the construction [$_{VP}$ only VP] is:

**(130)** $\lambda x \forall P[[P(x) \ \& \ P \in C] \ \longrightarrow \ P = {}^{\wedge}VP']$

where $VP'$ is the property that is the semantic representation of VP, and $C$ is the p-set of $VP'$, that is, a set of properties "like" $VP'$.

The $\lambda$-operator in the above can be viewed as an IOU for a suitable argument $x$ that is required in order to make (130) into a well-formed $\lambda$-free expression. In other words, (130) says that, given an appropriate argument $a$, one that a property "like" $VP'$ could take (hence one that $VP'$ could take), [$_{VP}$ only VP] plus argument translates as

**(131)** $\forall P[[P(a) \,\&\, P \in C] \longrightarrow P = {}^{\wedge}VP']$

For instance, the VP (132.1) is translated as the $\lambda$-expression (132.3), which is equivalent to (132.4), the p-set of the VP being (132.2), that is, the set of properties of washing something.

**(132)**
1. only [$_{VP}$ washed the ⟨DOG⟩]
2. $\{Q : \exists y[Q = {}^{\wedge}\text{wash}'(y)]\}$
3. $\lambda x \forall P[[P(x) \,\&\, P \in \{Q : \exists y[Q = {}^{\wedge}\text{wash}'(y)]\}] \longrightarrow P = {}^{\wedge}\text{wash}'(\text{dog})]$
4. $\lambda x \forall P[[P(x) \,\&\, \exists y[P = {}^{\wedge}\text{wash}'(y)]] \longrightarrow P = {}^{\wedge}\text{wash}'(\text{dog})]$

Note that wash$'$ is defined in such a way that wash$'(y)$ and wash$'$(dog) are themselves predicates that could take (say) Ross as an argument. When the VP (132.1) is combined with a subject NP to form a sentence, the translation of the NP becomes an argument to the $\lambda$-expression (132.4), producing a proposition, the translation of the sentence, by the process of $\lambda$-reduction. For instance, if the subject NP is Ross, applying (132.4) to the argument $x$ = Ross produces the expression (133.2). This proposition is Rooth's translation for the sentence (133.1). The expression can be glossed as (133.3).

**(133)**
1. [$_{NP}$ Ross] [$_{VP}$ only washed the ⟨DOG⟩].
2. $\forall P[[P(\text{Ross}) \,\&\, \exists y\ P = {}^{\wedge}\text{wash}'(y)\}] \longrightarrow P = {}^{\wedge}\text{wash}'(\text{dog})]$
3. *Any property that is true of Ross and that is a property of washing something is the property of washing the dog.* Or, informally, *if Ross washed something, it was the dog.*

Two notable features of the expression (133.2) are its lack of $\lambda$s and its use of intensions. The expression is the translation of a sentence, so it is a proposition, not a property like (132.4). Since the VP is being represented essentially as a proposition from which an argument is missing, the latter expression requires a $\lambda$-operator to act as an IOU for the missing argument. It is important to realize that the $\lambda$-operator is needed in the intermediate expression but *not*, however, in the expression that is the final goal of semantic interpretation, namely the translation (133.2) of the entire sentence which contains *only*.

The appearance of intensions in (133.2) is a result of their occurrence in the VP expression (132.4), where they are used in the representation of properties. Sub-expressions of (132.4) such as $P$, $\{Q : \exists y[Q = {}^{\wedge}\text{wash}'(y)]\}$, and ${}^{\wedge}$wash$'$(dog) are the expressions, all of the same type as a VP, associated with the VP *washed the dog* that *only* operates on.

Since (132.4) is an expression "about VPs", it is natural that it employs the concept of "properties", in the form of intensions.

Now Rooth develops alternate translations for *even* and *only* that talk about propositions rather than properties. Rooth employs the new translation for *only* to extend his basic semantics to cover cases in which it is adjacent to categories other than VP, particularly S and NP. He defines a basic sentential operator for *only*, $F_{only}$, which operates on sentence semantic forms, expressions of type $t$ in his Intensional Logic formalism. Then, one operator for each different semantic type of phrase that *only* may take as an argument is defined in terms of this basic operator (Rooth, 1985, pp. 120–121):

**(134)** *Rooth's* only *operators:*

1. $F_{only}(a) \;\equiv\; \forall P[[{}^{\vee}P \;\&\; P \in C] \longrightarrow P = {}^{\wedge}a]$
   where $a$ is an IL expression of type $t$, the type of a sentence
2. $F_{only,t} = F_{only}$
   (according to Rooth, *only* never takes this meaning in English)
3. $F_{only,\langle e,t\rangle}(b) \;\equiv\; \lambda x F_{only}(b(x))$
   where $b$ is an IL expression of type ⟨e,t⟩, the type of a VP
4. $F_{only,\langle\langle s,\langle e,t\rangle\rangle,t\rangle}(c) \;\equiv\; \lambda P F_{only}(c(P))$
   where $c$ is an IL expression of type ⟨⟨s,⟨e,t⟩⟩,t⟩, the type of an NP.

Similar operators are defined for *even*.

For example, to translate the VP (132.1), which is of type ⟨e,t⟩, the operator $F_{only,\langle e,t\rangle}$ is applied to the expression wash′(dog), which is the translation of the VP argument of *only*, yielding (135.1), which expands to (135.2).

**(135)** 1. $\lambda x F_{only,\langle e,t\rangle}(\text{wash}'(\text{dog})(x))$

2. $\lambda x \forall P[[{}^{\vee}P \;\&\; P \in C] \longrightarrow P = {}^{\wedge}\text{wash}'(\text{dog})(x)]$

As in (132.4) the translation of the VP is a $\lambda$-expression. The sentence in which the VP occurs is translated by applying (135.2) to the translation of the subject NP. If Ross is the subject NP, then the sentence is translated as

**(136)** $\forall P[[{}^{\vee}P \;\&\; P \in C] \longrightarrow P = {}^{\wedge}\text{wash}'(\text{dog})(\text{Ross})]$.

Unlike Rooth's original translation scheme, it appears that the cross-categorial translation outlined above lets the p-set $C$ remain unbound in the intermediate expressions. Since $C$ has the type of a set of propositions (in the expressions representing the VP and the sentence), it can only be understood as the p-set of a sentence. This p-set only becomes available during the final step of semantic interpretation at which the proposition representing the sentence is constructed. In our example, this p-set is (137.1), so the expression representing the sentence becomes (137.2), which can be glossed as (137.3).

**(137)**
1. $\{P : \exists y\ P = {}^\wedge\text{wash}'(y)(\text{Ross})\}$
2. $\forall P[[{}^\vee P\ \&\ \exists y\ P = {}^\wedge\text{wash}'(y)(\text{Ross})] \longrightarrow P = {}^\wedge\text{wash}'(\text{dog})(\text{Ross})]$
3. ***Any proposition that is true and that says Ross washed something is the proposition that Ross washed the dog.***

Either of Rooth's two methods of translating (133.1), with the glosses given as (133.3) and (137.3), captures the meaning of the sentence. The difference between (133.2) and (137.2) is that the former talks about properties, whereas the latter contains sub-expressions that are propositions. Propositions such as $^\wedge$wash′(dog)(Ross) that appear in (137.2) *contain* properties such as $^\wedge$wash′(dog) that are mentioned in (133.2). The expression constructed using Rooth's cross-categorial operator shares with (133.2) its lack of $\lambda$s and its use of intensions. (137.2) contains the intensional and extensional operators, but in this case their usefulness is to distinguish the description (intension) of a proposition from its truth value (extension) in a world. Note that in a semantics without intensions, such as Absity, these two types of object are not distinguished.

An aspect of Rooth's cross-categorial rules that turns out to be most useful is that they treat *only* and *even* basically as sentential operators, each taking an expression of type $t$ (i.e. a proposition) as an argument and producing another expression of the same type. Furthermore, Rooth's operators $F_{only,\langle e,t\rangle}$ and $F_{only,\langle\langle s,\langle e,t\rangle\rangle,t\rangle}$, which do not operate on translations of sentences, are derived from the basic, *sentential* operators. The translation rules (134.3) and (134.4) for *only* fronting a VP and NP respectively treat the resulting phrase as a quantification over propositions that are not fully developed: they are missing arguments. The $\lambda$-operator can be seen as an instruction to find an argument and to insert it in the appropriate place in the under-developed propositions. This has the flavor of categorial grammar: a VP with *only* is treated as a sentence missing its subject; an NP is a sentence missing its VP.

For instance, Rooth constructs $F_{only,\langle e,t\rangle}$ as $\lambda x F_{only}(b(x))$ by replacing the argument $a$ in the definition of $F_{only}$ with an under-developed proposition $b(x)$. An IOU, $\lambda x$, is then provided for the yet-unavailable argument $x$. When this argument becomes available (in the form of the translation of a subject NP), the mechanism of $\lambda$-reduction allows the expression generated by $F_{only,\langle e,t\rangle}$ to insert the missing argument such that $b(x)$ becomes a fully developed proposition wherever it occurs inside $F_{only}(b(x))$.

What's interesting about this procedure is that Rooth's non-basic operators want to talk about propositions—sub-expressions of $F_{only}$, and hence of all the *only* operators, are propositions. However, the arguments that the non-basic operators have available to them are not sentential expressions but expressions standing for NPs and VPs. The operators therefore produce $\lambda$-expressions that build up these expressions, until they become propositions, by providing them with missing parts. So when a $\lambda$-expression such as (135.2) produced by the operator $F_{only,\langle e,t\rangle}$ combines with the translation NP′ of a subject NP, the contribution of NP′ is exclusively to develop some sub-expression of (135.2), (wash′(dog)($x$))

in our example, into a fully-developed proposition (in the sense that it contains no unbound variable).

Rooth's approach of treating *even* and *only* as operators over sentences (actually, over expressions of type $t$) suggests a simpler method of translating *even* and *only* sentences compositionally *without* using $\lambda$-expressions in intermediate representations.

My approach is that there will be a single *only* operator, a sentential operator, essentially Rooth's $F_{only}$; similarly for other focusing subjuncts. This operator takes the semantic representation of a sentence as an argument and produces another semantic representation of the same type. When sentential objects are not available, as in the interpretation of [$_{VP}$ only VP], I delay the application of the operator until such a point as fully developed propositions, the semantic objects of sentences, *are* available. These points are those at which sentences are built, namely (in GPSG), the ID rules to build S. Thus, this operator is only used by the semantic rules that correspond to syntactic rules for building *sentences*.

The essential difference between Rooth's cross-categorial rules and the rule I propose is the manner in which the expressions they produce combine with the rest of the sentence being translated. Rooth's expression (135.2) for the VP *only washed the dog* combines with the translation of the subject NP *Ross* by using the latter to "develop" certain sub-expressions of (135.2) into propositions: specifically, wash′(dog)($x$) develops into wash′(dog)(Ross). I propose, on the other hand, that [$_{VP}$ *only* VP] be represented simply as the translation of VP along with a notation that the *only* operator is **latent**, and should be applied at the sentence level to obtain the final translation for the sentence containing [$_{VP}$ *only* VP]. When [$_{VP}$ *only* VP] is combined with its argument, the subject NP, the argument develops the VP into a full-fledged proposition, rather than developing a $\lambda$-expression into a quantification over propositions. Then, an expression having sentential type being available, the sentential *only* operator is applied to the proposition, producing the same quantifier over propositions as Rooth's cross-categorial translation.

Figure 4.4 illustrates that the two methods of translating *only*, the first using $\lambda$-expressions and the second using the delayed application of an operator, ultimately produce the same translation (modulo intensions) of a sentence in which *only* is adjoined to a verb phrase. Pictorially, the idea is that the diagram in Figure 4.5 commutes.

The approach outlined above easily handles *only* adjoined to any kind of phrase, not just VP. The *only* operator itself does not manipulate the actual adjunct of *only*, but comes into play at the level of the construction of the representation of the sentence. Any constituent [$_{XP}$ *only* XP] is represented just the same way as XP is, but with the latent *only* operator marked. When the semantic representation of a sentence marked with this latent operator is produced, the operator is applied to the representation otherwise obtained for the sentence. Operators similar to the sentential *only* operator are defined for other focusing subjuncts. More than just indicating the presence of a focusing subjunct, the fs feature introduced in Section 4.1 represents the latent, delayed operator. In effect, then, fs is a semantic feature, and it represents the semantic content of a focusing subjunct.

| Constituent | Translation per Rooth | $\lambda$-free translation |
|---|---|---|
| VP | $VP'$ | $VP'$ |
| $[_{VP}$ *only* VP] | $\lambda x \forall P[[{}^{\vee}P \,\&\, P \in C]$<br>$\longrightarrow P = {}^{\wedge}VP'(x)]$ | <br>$VP'$ + delayed operator $o$ |
| $[_S$ NP [ *only* VP]] | $(\lambda x \forall P[[{}^{\vee}P \,\&\, P \in C]$<br>$\longrightarrow P = {}^{\wedge}VP'(x)])(NP')$<br>$= \forall P[[{}^{\vee}P \,\&\, P \in C]$<br>$\longrightarrow P = {}^{\wedge}VP'(NP')]$ | $VP'(NP')$ + delayed op. $o$<br>$= o(VP'(NP'))$<br>$= \forall P[P \,\&\, P \in C$<br>$\longrightarrow P = VP'(NP')]$ |

Figure 4.4: Comparison of two methods of translating *only*

$$\begin{tikzcd}
VP' \arrow[r, "F_{only,(e\ t)}"] \arrow[d, "\text{consume } NP'"'] \arrow[dr, phantom, "\text{Rooth}" near start] \arrow[dr, phantom, "\text{Lyons}" near end] & \lambda x F_{only}(VP'(x)) \arrow[d, "\text{consume } NP'"] \\
VP'(NP') \arrow[r, "F_{only}(VP'(NP'))"'] & F_{only}(VP'(NP'))
\end{tikzcd}$$

Figure 4.5: This diagram commutes

Rooth's families of cross-categorial *even* and *only* operators could presumably be extended to handle any number of phrase types, not only VP and NP. But this would require the definition of a new operator in each family and for each different type of constituent allowed to adjoin to *even* and *only*. It is an advantage of my approach that only a single operator need be defined for each focusing subjunct. This is made possible by defining only a sentential operator for each focusing subjunct, and by postponing the application of this operator to the semantic rules to construct sentences. Several pieces of evidence motivate this approach:

- Rooth recognizes that, in order to define a family of cross-categorial operators for (say) *only*, a basic operator must be defined that operates on an expression of sentential type.

- Rooth's *only* and *even* operators all employ a variable $C$ that is bound, in the expression representing a full sentence, to the p-set of the *sentence*.

- Focusing subjuncts normally occur at most once per *sentence*. Even granting the acceptability of sentences containing several focusing subjuncts, such sentences are clearly semantically complicated.

The principal advantage of my approach is that it constructs essentially the same final translation of a sentence as Rooth's, but avoids using the $\lambda$-operator. The system is still compositional; well-formed semantic objects are still constructed for each well-formed syntactic phrase, including those with focusing subjuncts. It just happens that these objects are just those constructed for the phrase to which the focusing subjunct is adjoined, enhanced with a latent operator. In a sense this constitutes a bit of a cheat: the latent operator is described by name (e.g. only), but it is really a function. If it were necessary to represent arbitrary functions of the type of the sentential *only* and *even* operators, the expressive power of $\lambda$-expressions would be required, and the representation would become more intractable.

Fortunately, for several reasons, there is no reason to believe that just because focusing subjuncts need to be represented as operators, a semantics for English requires the representation of arbitrary functions. Note that the *only* and *even* operators I define only operate at the sentence level, and thus call for special handling only by the few semantic rules that construct representations of sentences. It is not a general feature of my semantics that semantic rules all use operators, as is the case in McCord's system. Note also that the operators defined for *only* and *even* can be used for the semantic interpretation of all of the focusing subjuncts. Thus they are in some sense "basic" to English, and it is justifiable to make them part of a system of semantics without bringing in the representation of functions in general, which would make the semantics highly intractable.

The semantics I define explicitly in Section 4.2.2 omits intensions. Though intensions are an important concept in language, and a representation of intensions may be necessary to capture the full expressiveness of English, I claim that they are not crucial to the understanding of the meaning of focusing subjuncts. Thus no complications can be expected to arise from the incorporation of my semantics for focusing subjuncts into a scheme in which intensions *are* represented. Furthermore, in my method of constructing translations of sentences, the notion of "properties" is not as central as in Rooth's original (non-cross-categorial) translation rules for *only*, making intensions less important.

### 4.2.2 Details of the semantics

Three semantic objects are computed for and attached to each syntactic constituent, in parallel with the syntactic processing. The objects are of the types defined in my Absity-like frame representation. They are attached to a node as values on the following features:

Assert: The asserted meaning of the constituent, its contribution to the sentence's *asserted meaning.*

Presupp: The constituent's contribution to the sentence's *presupposition* (actually to the category of meaning to which the *non-asserted meanings* of focusing subjuncts belong, whatever it may be).

P-set: A prototype of the semantic objects in the node's p-set. All objects matching this object are in the node's p-set.

In order not to commit myself to the status of the non-asserted meanings of focusing subjuncts, I will reserve the presuppositional feature for use solely to represent these meanings, which are introduced by the focusing subjunct sentential operators. I will not prejudice my neutrality, nor complicate the present exposition, by incorporating, say, the existential presupposition introduced by an NP into the NP's presupp feature.

To define my semantics of focusing subjuncts is to define the calculation of the values on each node of the semantic features, that is, to define the semantic rules corresponding to each syntactic rule. The semantics makes use of the three features described above, as well as the focus and fs features (but primarily as input to compute a node's semantics).

**Computing the assert feature value** The assert feature is just the semantic object that Absity would construct for a constituent from its daughters' assert feature values. Figure 4.6 shows the rules for computing this value. Each of these rules corresponds to a syntactic rule of the illustrative grammar.

**Computing the presupp feature value** For all ID rules but sentence rules, the presupp feature on the parent node is undefined. As stated above, I reserve this feature for the non-asserted meanings introduced by focusing subjunct operators.

**Computing the p-set feature value** Two cases are distinguished:

Case 1: If the parent node X (being constructed) is [+ focus], its p-set is a variable of the same type as the assert object, as per Rooth (see Section 2.5).

Case 2: Otherwise, the p-set of X is constructed in a manner exactly paralleling the construction of the assert feature, except that the p-set values of the constituent phrases, rather than their assert values, are used as inputs. Note that a second-stage semantic rule (see below) may affect the construction of the p-set of a sentence (rules 101 and 207 in the illustrative grammar).

**Application of the focusing subjunct operators** The semantics given for sentence rules such as 101 and 207 in the illustrative grammar is incomplete. The semantic features obtained for S in the way described above are the output only of *first*-stage semantic rules. Second-stage rules apply for the computation of the semantic feature values of sentences.

**(138)** *Second-stage semantic rule for S:*

1. Output of first stage:
   S[⟨assert $\alpha$⟩, ⟨p-set $\beta$⟩, ⟨fs $\gamma$⟩]

| Rule number | Daughter features | Semantic rule |
|---|---|---|
| 101 | XP$^a$ = (agent = $\alpha$)<br>H$^a$ = (frame $\beta$ sf-pairs) | S$^a$ = (frame $\beta$ (agent = $\alpha$) sf-pairs)* |
| 1 | H$^a$ = (frame ?$\alpha$) | VP$^a$ = (frame ?$\alpha$) |
| 2 | H$^a$ = (frame ?$\alpha$)<br>PP[obj]$^a$ = (slot$\beta$ = $\beta$) | VP$^a$ = (frame ?$\alpha$ (slot$\beta$ = $\beta$)) |
| 3 | H$^a$ = (frame ?$\alpha$)<br>PP[obj]$^a$ = (slot$\beta$ = $\beta$)<br>PP[to]$^a$ = (slot$\gamma$ = $\gamma$) | VP$^a$ = (frame ?$\alpha$ (slot$\beta$ = $\beta$) (slot$\gamma$ = $\gamma$)) |
| 5 | H$^a$ = (frame ?$\alpha$)<br>PP[iobj]$^a$ = (slot$\beta$ = $\beta$)<br>PP[obj]$^a$ = (slot$\gamma$ = $\gamma$) | VP$^a$ = (frame ?$\alpha$ (slot$\beta$ = $\beta$) (slot$\gamma$ = $\gamma$)) |
| 131 | H$^a$ = WH† | NP$^a$ = WH |
| 130 | | NP$^a$ = H$^a$ |
| 30a | Det$^a$ = (Det ?$\alpha$)<br>H$^a$ = (frame ?$\alpha$ sf-pairs) | NP$^a$ = (Det ?$\alpha$ (frame ?$\alpha$ sf-pairs)) |
| 132 | Det$^a$ = (Det ?$\alpha$)<br>H$^a$ = (frame ?$\alpha$ sf-pairs)<br>S[+R]$^a$ = $\beta$‡ | NP$^a$ = (Det ?$\alpha$ (frame ?$\alpha$ sf-pairs)) + $\beta$ |
| 30 | | N1$^a$ = H$^a$ |
| 38 | H$^a$ = slot$\alpha$<br>NP$^a$ = $\beta$‡ | PP$^a$ = (slot$\alpha$ = $\beta$) |
| 107 | H$^a$ = (slot$\alpha$ = $\alpha$)<br>VP[BSE] = (frame ?$\beta$) | VP[+AUX]$^a$ = (frame ?$\beta$ (slot$\alpha$ = $\alpha$)) |
| 207 | H$^a$ = (slot$\alpha$ = $\alpha$)<br>VP[BSE] = (frame ?$\beta$)<br>PP[subj]$^a$ = (slot$\gamma$ = $\gamma$) | VP[+INV]$^a$ =<br>(frame ?$\beta$ (slot$\alpha$ = $\alpha$) (slot$\gamma$ = $\gamma$))* |
| 111 | | XP$^a$ = H$^a$ |

In all the rules, X$^a$ is the value of the assert feature of X.

* First stage; see below.

† *who* or *what.*

‡ $\beta$ is an instance or (possibly, in rule 132) a frame implication (see text).

Figure 4.6: Semantic rules for the assert feature

2. Output of second stage:
   - If $\gamma =$ "–" then
     S[⟨assert $\alpha$⟩, ⟨p-set $\beta$⟩]
   - else
     S[⟨assert op$1_\gamma(\alpha,\beta)$⟩, ⟨presupp op$2_\gamma(\alpha,\beta)$⟩, ⟨p-set $\beta$⟩]

Op$1_\gamma$ and op$2_\gamma$ are the outputs of the sentential operator for the focusing subjunct $\gamma$.

A cleaner separation between syntax and semantics may be obtained by not having both the syntax and the semantic rules modifying the fs feature. This may be accomplished by adding a syntactic rule whose sole purpose is to support of the application of a sentential operator:

(139) 201 S $\longrightarrow$ H[fs $\alpha$]

The FFP, part of syntax, causes ⟨fs $\alpha$⟩ not to percolate to S. Then, the first-stage semantics described above is made the complete semantics for the old S rules (101 and 207 in the illustrative grammar). The second-stage semantics given in (138) becomes the semantics for the ID rule (139). Finally, S[fs $\alpha$] is specified as a non-initial category in the grammar, if $\alpha \neq$ "–", forcing (139) to be used to generate a well-formed sentence containing a focusing subjunct.

**The sentential operators** The sentential operators for *only* and *even* are given below. The one for *too* is the same as that for *even*. Those for the other focusing subjuncts are similar.

(140)
1. op$1_{only}$(A, P) = if P then A
2. op$2_{only}$(A, P) = A
3. op$1_{even}$(A, P) = A
4. op$2_{even}$( (the $?x$ frame-descrA), (the $?y$ frame-descrP) )
   = (anew $?y \neq ?x$ (frame-descrP))

If P then A is intended as a directive to the underlying knowledge base to insert the rule that any frame matching P is just the frame A, that is, A is the unique frame matching P. This is just the frame version of Rooth's operator $F_{only}$ given in (134). I will call this directive a **frame implication.** It is similar in character to a frame determiner, in that it is a function that manipulates the underlying knowledge base. A separate frame implication is postulated for questions containing focusing subjuncts (operators taking frame statements with the frame determiner (question $?x$) would be required).

The form (anew $?y \neq ?x$ frame-descrP) is also a new type of entity in the semantics. I shall treat it as a frame determiner. It is the frame version of Rooth's $F_{even}$ (Rooth, 1985, p. 120). It is a directive to the knowledge base to retrieve or create a frame instance,

$?y$, matching frame-descrP, that is *not* the frame instance identified by the label/variable $?x$. Like the frame determiner (the $?x$), such a frame instance $?y$ should be inserted if not already present in the knowledge base. It is assured that $?x$ corresponds to an inserted instance by virtue of its being the label of the frame statement produced by (140.3).

The reader may be disturbed that I have abandoned strong typing by allowing sentences to be one of two types: frame instances and frame implications. He or she may further wonder what the consequences of this are for compositionality. Indeed, the semantics of phrases in which a sentence is embedded can no longer rely on the sentence's semantic object being of type *instance*. A frame statement is meaningful as a slot filler, because its knowledge base semantics makes it equivalent to an instance (the frame determiner the, at least, is a function that retrieves an instance from the knowledge base). But a frame implication is a function that asserts a rule, not one that returns an instance.

The following reply is proffered. Embedded sentences contribute semantically in one of two ways. First, they can contribute as frames independent of the matrix sentence, except that they are linked to it by "pointer" labels. For example, Hirst translates (141.1) as (141.2) (1987, p. 49).

(141) 1. the grey cat that Ross loves

2. (the $?x$ (cat $?x$ (color = grey)
   (a $?y$ (love $?y$ (agent = Ross) (patient = $?x$)))))

The label $?x$ is the link between the two frames. It is unproblematic in a case such as this that the embedded sentence be represented by an independent frame-implication rather than a frame instance.

In the second case, an embedded sentence contributes semantically as an instance that fills a slot of some verb taking S arguments, such as a modal verb or the verb *to be*.

(142) 1. John believes that Sue loves Bill

2. (the $?x$ (believe $?x$ (experiencer = John)
   (patient = (a $?y$ (love $?y$ (agent = Sue) (patient = Bill))))))

For modal verbs and for predication, semantic representation is in any case problematic without dual sentence typing. A solution whereby belief is handled by having a partition of the knowledge base representing an agent's belief context is possible.[1] A belief such as that in (142) would appear in the knowledge base by inserting into the partition representing John's beliefs the frame statement that is the patient of the belief frame in (142.2). It doesn't matter whether we manipulate John's partition with a frame determiner operation or a frame implication. So in the second case, like the first, allowing the embedded S to belong to one of several semantic types adds no complications for semantic representation

[1] Knowing about belief contexts may in any case be necessary to handle presuppositions; see (Horton and Hirst, 1988)

that were not already there. As an aside, it is interesting to note that when an *only* sentence is embedded in a belief report, it is negative information that is believed. For example, in (143) it is the negative information that Sue doesn't love anyone else that we would say John found out about.

(143) John found out that Sue only loves ⟨BILL⟩.

As a frame representation of predication is not available in Absity, I am unable to reconcile my departure from strong typing with cases in which a predication takes an embedded S argument (usually as its subject). Hirst (1987, p. 65) points out that in any case the representation of predication in a frame representation may require that two different semantic types exist for sentences.

I feel I have justified my contention that dual semantic typing for sentences is simply the cost of being able, in a frame system, to represent the semantics of certain facets of language.

## 4.3 Frame representation semantics

My semantics would be incomplete if it just defined a mapping from one kind of linguistic representation (written English) to another (a frame representation system). It is necessary to also define the semantics of the frame representation, especially those of the new types of objects I introduce to handle focusing subjuncts.

Following Hirst (1987), the semantics of my representation is defined in terms of a knowledge base representing the world. The semantics for sentential objects (in Absity, frame statements) is provided in the form of definitions of their effect when they are passed to the knowledge base for resolution. In Absity, this involves defining the frame determiners as knowledge base functions:

> A frame determiner is a function that, like **assert** and **retrieve**, adds or gets formulas from the database but, unlike those basic functions, takes into account notions of *context* and *focus* (Hirst, 1987, p. 17).

For the sake of simplicity, the notions of "context" and "focus" will be waved away by taking all of the information in the knowledge base to be "in focus" and "in context". Hirst assumes that there is a model of focus,[2] such as that of Sidner (1979), that could be plugged in to his semantic interpreter and used to retrieve discourse entities that are "in focus". These are the entities that at a given point in the discourse are (somehow) rapidly found by language users. In principle I could also assume the existence of a plug-in model of this type of focus.

Recall that a frame statement is made up of a frame determiner plus a frame description. When Absity finishes building a frame statement and passes it on for processing to

[2] This is a different meaning of *focus* than the one I use extensively in this thesis.

the knowledge base, it invokes the frame determiner (as a knowledge base function) with the frame description as data. Hirst informally provides the semantics for three frame determiners (1987, pp. 17–18):

> The two main frame determiners are the and a, and their semantics are what their names suggest. The the function is like retrieve, except that before searching the database it first looks in the focus for a matching instance that is *active* in context (Hirst (1981a), (1981b)). (It is an error if there is not either exactly one match in focus, or none in focus but exactly one in the database) ...The a functions asserts the existence of a new frame instance (and returns as a value its name), but allows the possibility that it may turn out to be identical to a pre-existing instance; in particular, if there are frame instances of the same type in focus, a will assume that its argument is identical to one of them ...The frame determiner question is used in questions. It is very close to a simple knowledge-base retrieval function, looking for an instance that matches its argument and not considering focus at all. It returns a list of matching instances, and if its argument included any free variables, it also gives the bindings of those variables for each instance it found.

I attribute the same semantics to these frame determiners.

The knowledge base semantics assigned to the anew frame determiner is similar to that of the frame determiner a. However, this frame determiner takes two arguments: a handle $?x$ onto a frame instance assumed already to be in the knowledge base, and a frame description. It asserts the existence of a new frame instance that matches the frame description but is *distinct* from $?x$.

For example, suppose the sentence

(144) Even Ross washed the dog.

has as its assertion and presupposition, respectively, the frame representations (145.1) and (145.2).

(145) 1. (the $?y$ (wash $?y$ (agent = Ross) (patient = dog)))

2. (anew $?x \neq ?y$ (wash $?x$ (patient = dog)))

When processing (145.2) in the knowledge base, $?y$ is assumed to identify the frame (146) that is in the knowledge base as a result of invoking (145.1) as a knowledge-base function call:

(146) (wash $?y$ (agent = Ross) (patient = dog))

The meaning of (145.2) can then be glossed as "there is a dog-washing frame besides the one having Ross as its agent". The meaning of the frame determiner anew as a knowledge-base function is that a new frame matching the frame descriptor (wash $?x$ (patient = dog)) is added to the knowledge base, if one is not already present *other* than the one identified as $?y$, namely (146). The frame determiner function anew, like the function a, returns a matching frame (one matching (wash $?x$ (patient = dog)) in the above example) if one is already present in the knowledge base. It fails if such a frame is not present and cannot be added (for whatever reason).

Note that if a new frame is added to the knowledge base, the information that the agent of (wash $?x$) is not Ross is *not* attached to it. This cannot be done in a semantics that is based on Rooth's, since, for reasons discussed in Section 2.5, the focus *Ross* of the *even* in (144) is not an argument of the focusing subjunct. It is not available as a distinct semantic object to the semantic interpreter at the point at which the semantic objects (145.2) and (145.1) of the sentence are constructed.

I have some reservations about the appropriateness of adding a frame to the knowledge base for "non-asserted" meanings, such as (145) in the case of sentence (144). It may be preferable to process frame statements differently when they represent the non-asserted meanings introduced by focusing subjuncts from when they represent assertions of sentences. This can be accomplished by defining a "presuppositional" version of anew that merely verifies the presence of a frame in the knowledge base or checks that the appropriate frame *can* consistently be added to the knowledge base, without actually adding it. Better, it can be accomplished without re-defining anew by causing all frames added due to non-asserted meaning to be segregated in a separate partition of the knowledge base. This is a reasonable thing to do, especially if presuppositions are treated as beliefs of the participants in a discourse. This partition could be subject to rules different from those that apply to the rest of the knowledge base; for example, frames in it could be retracted if they conflict with later assertions. As a simplification, I will not take either of these steps in this thesis.

A retrieval version of anew for yes/no questions like (147) is easy to define.

(147) Did even ROSS wash the dog?

The frame statement (qanew $?x \neq ?y$ frame-description) succeeds exactly when a frame $?x$ matching frame-description is present in the knowledge base and is not the frame $?y$. It returns the frame $?x$. qanew does not add a frame to the knowledge base. For example, (147) may "assert" (148.1) and presuppose (148.2).

(148) 1. (question $?y$ (wash $?y$ (agent = Ross) (patient = dog))).

2. (qanew $?x \neq ?y$ (wash $?x$ (patient = dog))).

(148.2) succeeds if there is a frame (wash $?x$ (patient = dog)) in the knowledge base other than (wash $?y$ (agent = Ross) (patient = dog)).

The semantics of frame implications is a bit more complicated. A frame implication if A then P behaves as a two-argument knowledge-base function. It is essentially an insertion function: but what it inserts into the knowledge base is a rule to the effect that a frame only matches A in the database if it matches P. P is expected to already be inserted in the knowledge base (an *only* sentence, for example, inserts P as a presupposition when it asserts if A then P). Before the rule is inserted, the knowledge base manager verifies that there is no frame matching A that does not match P. Any subsequent attempt to assert or retrieve a frame matching A but not P fails. It is this last fact that allows frame implications to support the *negative information* that *only* sentences seem to convey.

For example, the sentence (149.1) asserts the frame implication (149.2).

(149) 1. Ross only washed the DOG.

2. (if (wash ?$y$ (agent = Ross))
then (wash ?$y$ (agent = Ross) (patient = dog)))

The processing of this frame implication results in the insertion of a rule in the knowledge base to the effect that the only wash frame in the knowledge base with Ross as its agent has dog as its patient. Subsequently, no wash frame with the same agent can be inserted having a different patient (a consistency check on subsequent insertions may be required in order to implement this). The insertion of the rule fails if there is already such a frame in the knowledge base. Ideally, the knowledge base should have a concept of time, so that this consistency check applies only to those wash frames occurring at or during some specified time. Otherwise, after (149.1) is processed, Ross can never wash anything again. I do not deal with the representation of time in this thesis.

A retrieval version of frame implication appears necessary for handling questions and perhaps relative clauses. For example, in interpreting a phrase such as

(150) the grey cat that only Ross loves

what is required is not to *insert* a frame implication, but to retrieve a frame implication (in order to find out what it is that only Ross loves). Something similar is needed in order to accomplish a variable instantiation of the sort done by Absity's question frame determiner, in order to answer wh-questions such as

(151) Who does only Ross love?

The desired knowledge base semantics for this question would be the retrieval of a frame

(152) (love ?$x$ (agent = Ross) (patient = WH))

and the instantiation of the variable WH to the appropriate value, so that the rule

(153) (if (wash ?$y$ (agent = Ross))
then (wash ?$y$ (agent = Ross) (patient = WH)))

be true of the database. This semantics would amount to the retrieval of some object that Ross loves but nobody else does (negative information) under a closed-world assumption that the database contains all true (positive) instances. I propose the construct

(154) qif A then P

which succeeds as a knowledge base function call exactly when there is a unique frame matched by the frame description P that is also matched by the frame description A. If it succeeds, bindings for any variables WH in A are returned as well. Put differently, qif A then P succeeds just when the rules arising from the frame implication if A then P *could* consistently be added to the knowledge base.

This construct can also be used to answer yes/no questions like

(155) Did Ross only wash the DOG?

There are two ways of showing that the answer to such a question is *yes*. Firstly, if the sentence (149.1) has been interpreted, there will be a rule in the knowledge base as a result of the frame implication (149.2). The second way is to prove that such a rule *could* consistently be added to the knowledge base, that is, the frame implication (149.2) would succeed as a knowledge base function call. This is the same as the condition for the success of qif A then P. In the case of the question (155) this amounts to proving that Ross didn't wash anything else. Of course, we can only prove this if our knowledge base contains negative information, or if we make a closed world assumption that all positive information is present in the knowledge base. Unfortunately, it does not appear that the inference engine required for the proof of qif A then P is trivial to implement.

The question of the suitability of a frame implication for the purposes of composition is dealt with by the comments at the end of the previous section, 4.2.2.

# Chapter 5

# The implementation

IDEO[1] is a semantic interpreter that incorporates the semantics for *even* and *only* described in Chapter 4. The implementation is in Edinburgh C-Prolog, running under UNIX (BSD 4.3) on a Sun-4 computer.

The semantic interpreter is not particularly robust, extensive in its coverage, or efficient at interpreting the focusing subjunct constructs for which it was designed. Its purpose is solely to illustrate the feasibility of implementing the semantics of focusing subjuncts developed in previous chapters.

The interpreter is written with simplicity and transparency in mind. The GPSG parsing mechanism is not hidden or "compiled in" to a DCG or an ATN: the GPSG principles are clearly visible at the top levels of the program. Semantic rules are distinct from syntax rules, but operate in tandem with them.

## 5.1 The grammar

IDEO's parser is a top-down, straightforward implementation of GPSG. It takes the naïve approach to parsing suggested by the theory itself. A phrase is parsed by recursively parsing its constituents, obtaining GPSG categories for each one. Categories for individual words are given in the lexicon. Then, a candidate category is non-deterministically constructed for the parent phrase. This category and the resulting parse tree are checked for conformity to the well-formedness conditions imposed by the principles of GPSG (see Section 3.3) and by the ID rules for the particular grammar being used. A complete parse consists of a tree each of whose nodes is a GPSG category, that is, a feature structure.

In fact, the parser is not wholly transparent with respect to all of the principles of GPSG. Because of the excessive non-determinism of the naïve approach being used, a bottom-up component is added to provide direction as to which ID rules may be applied to parse a phrase, given its starting word. The LP rules are compiled with the ID rules to yield DCG-

[1] Interpreter Designed for *Even* and *Only*

like phrase structure rules that I call **grammar rules** (see Section 5.1.2 below). The Head Feature Convention is partly compiled into these rules, so that some candidate parse trees in which the head features of a phrase and its head constituent do not match are never attempted. FSDs and FCRs are implemented to the extent that the grammar rules license only certain combinations of features. That is, not all of them are implemented and those that are "fall out" of the grammar rules.

The parser consists of several components, each of which implements some facet of GPSG and is described below:

- Grammar rules (Immediate Dominance rules and Linear Precedence statements)
- Control Agreement Principle
- Head Feature Convention
- Foot Feature Principle

### 5.1.1 Data structures

The central role of a GPSG parser is to build and manipulate categories. Categories are feature structures represented in IDEO by the Prolog functor `struct(SFList)`, where `SFList` is a Prolog list of slot–filler pairs. A slot–filler pair is a functor `slot(filler)`, where `filler` is either an atom or another category. For example, the NP category, {⟨N,+⟩, ⟨V,–⟩, ⟨BAR,2⟩} in GPSG, is stored internally as `struct([n(+), v(-), bar(2)])`. The complex category

(156) { ⟨N,+⟩, ⟨V,–⟩, ⟨BAR,0⟩ ⟨AGR,{⟨PERS 3⟩, ⟨NUM PL⟩}⟩ }

is stored as

(157) `struct([ n(+), v(-), bar(0), agr(struct([pers(3), num(pl)])) ]).`

Parse trees have no independent representation of their own. Instead, they are embedded in the feature structures of their nodes. That is, each of a node's daughter categories appears as a feature on the parent's category (as per Shieber (1986, pp. 21*ff*)). So does the grammar rule number used to construct the node. When a parse tree is displayed, it is this number, rather than the entire category (a very cumbersome structure indeed) that labels the node. This method of displaying parse tress is used by Evans (1985, p. 227). Figure 5.1 shows a sample category (much simplified) for the sentence *Ross sleeps.*

### 5.1.2 Grammar rules

IDEO's grammar rules are phrase-structure rules, most of them expressed in Prolog's DCG notation (Pereira and Shieber, 1987, p. 72). Therefore, they must be thought of in GPSG terms as Immediate Dominance rules to which Linear Precedence statements have been

```
struct([
   n(-), v(+), bar(2), subj(+),   agr(struct([pers(3), num(sing)])),
   np(struct([
      n(+), v(-), bar(2), agr(struct([pers(3), num(sing)])),
      head(struct([
         n(+), v(-), bar(0), subcat(130), agr(struct([pers(3), num(sing)])),
         form(Ross)
      ]))
   ])),
   head(struct([
      n(-), v(+), bar(2), agr(struct([pers(3), num(sing)])),
      head(struct([
         n(-), v(+), bar(0), subcat(1), agr(struct([pers(3), num(sing)])),
         form(sleeps)
      ]))
   ]))
])
```

Figure 5.1: Category for *Ross sleeps*

applied in order to produce an ordering of the dominated nodes. Note that the GPSG parser of Evans (1985) allows ID rules to be specified directly, but compiles them with the LP statements, the HFC, and so on, to produce phrase-structure rules.

Grammar rules make up the top level of the parser. They drive the parser, and themselves invoke the other components, which check well-formedness conditions on the categories and trees produced by the rules. It should be noted that invocation of the other components usually results in the modification of the categories generated by the rules. An under-specified category that is *consistent* with some principle is made more restricted (more specific) by the constraints imposed by the principle. Grammar rule number 30 shown in Figure 5.2 is typical, and invokes all of the components of the parser. The phrase-structure skeleton to this rule is

```
(158)  node(30, NP, NP7) ⟶
                 parse(123, struct([det(Det)]), D),
                 parse(121, H, H2).
```

By itself, this just says that grammar rule 30 can parse an input if rule 123 parses an initial substring of the input and rule 121 parses the remainder. Both node and parse are Prolog predicates to parse any phrase (the difference between them is explained below). The Prolog predicates in braces in the figure are the well-formedness conditions applying to the NP category and the tree it dominates, including specifications of the categories appearing in an ID rule (ID rule 30 in our example). The grammar rule itself therefore embodies the ID rule

```
1     node(30, NP, NP7) --->
2           {category(NP2, np)},
3           {unify(NP, NP2, NP3)},
4           parse(123, struct([det(Det)]), D),
6           parse(121, struct([n(+), v(-), subcat(30), bar(0)]), H2),
5           {cap(D, H, D2, H2)},
7           {hfc(NP3, H2, [subcat, bar], NP4)},
8           {ffp(NP4, [H2], [], NP5)},
9           {syntax(NP5, [rule(30), determiner(D2), head(H2)], NP6)},
10          {semantics(30, NP6, [D2, H2], NP7)}.
```

Figure 5.2: A typical grammar rule

(159) NP ⟶ Det, H[30]

(with the LP statement Det ≺ N compiled in), as will be explained in the line-by-line explanation of figure 5.2 that now follows.

- {category(NP2, np)}
  looks up np in a dictionary of short-forms for categories, and finds struct([n(+), v(-), bar(2)]). NP2 is instantiated to that value.

- {unify(NP, NP2, NP3)}
  means that NP3 is the most general feature structure subsumed by both NP and NP2. NP is the under-specified feature structure used as the starting point for constructing the NP's category. NP2 is instantiated to the np category; therefore rule 30 can only be used to build NPs, as desired.

Throughout the program, rather than using the Prolog unification mechanism in the manner of Evans (1985, p. 230), most of the Prolog predicates produce separate output categories that are formed by adding restrictions (values on features) to an input category. In the grammar rule of Figure 5.2, for example, the variables NP, NP2, NP3, ..., NP7 constitute a sequence of categories, each one more restricted than (subsumed by) the previous. They are each stages of construction of the NP's category: NP is a starting point, NP7 is the final category, and the other variables are used for intermediate categories generated along the way by the different parser components invoked by the grammar rule.

- parse(123, struct([det(Det)]), D)
  looks for a determiner as the first constituent of the NP.

The three-argument parse predicate is actually a front end to the three-place node predicate (grammar rule). parse contains the bottom-up component of the parser; it prevents the application of grammar rules that cannot be started by the next token in the input.

The first argument to parse is the grammar rule number used to parse the determiner constituent. In most cases parse is invoked with this argument being an uninstantiated variable. However, since only rule 123 (a lexical rule) can match a determiner, for the sake of efficiency I specify it in rule 30. The second and third arguments to `parse` are the initial category (based on top-down expectations) and final computed category, respectively, for the determiner node. The node is specified as being a determiner category by passing `parse` the category {⟨DET⟩} as its starting point.

- `parse(121, struct([n(+), v(-), subcat(30), bar(0)]), H2)`
  constructs the category for the second constituent of the NP.

The starting point for this construction is `struct([n(+), v(-), subcat(30), bar(0)])`. Top-down expectations impose the requirement that `H` be {⟨BAR, 0⟩, ⟨SUBCAT, 30⟩}, which is what H[30] means in the ID rule (159). In addition, `H` must be {⟨N,+⟩, ⟨V,−⟩}. This last requirement represents a pre-compilation into grammar rule 30 of the Head Feature Convention, which stipulates (among other things) that the N and V features on the NP category and its head must match; recall that in our example grammar rule, the category for the NP has `n(+)` and `v(-)` instantiated onto it. This pre-compilation is performed for the sake of reducing non-determinism in the parser.

- `{cap(D, H, D2, H2)}`
  invokes the Control Agreement Principle component of the parse (see below). The determiner (category `D`) controls the head (`H`). In the actual implementation, cap is invoked before the head is parsed, to give the parser direction as to the desired agreement features on the head.

- `{hfc(NP3, H2, [subcat, bar], NP4)}`
  checks that the tree headed by the NP node will conform to the Head Feature Convention.

`NP4` is the NP's category when restricted to conform to the HFC. The head features SUBCAT and BAR are specified as exempt from being identical on the NP and its head. Because C-Prolog is not fully declarative, and because a grammar rule operates on a sequence of input and output variables standing for the NP's category, I take care to invoke `hfc` after cap, so the the head feature `Agr` instantiated on the head by its determiner controller gets instantiated onto the NP's category.

- `{ffp(NP4, [H2], [], NP5)}`
  ensures that the tree headed by the NP node conforms to the Foot Feature Principle (see below). `NP5` is the resulting, possibly more restricted version of the category for the NP.

- `{syntax(NP5, [rule(30), determiner(D2), head(H2)], NP6)}`
  attaches parse-tree pointers to the NP's category. Pointers to its determiner and head categories are placed on its `determiner` and `head` features respectively. The rule number (i.e., 30) used to construct the NP is noted on the `rule` feature of `NP6`.

- {semantics(30, NP6, [D2, H2], NP7)}
computes the semantic features for the NP node from the semantics of its daughter categories D2 and H2. The parallel semantic rule of grammar rule 30, semantic rule number 30, is used for this computation (see Section 5.3).

NP7 is the final NP category produced by the grammar rule in Figure 5.2.

### 5.1.3 Head Feature Convention

IDEO's parser uses the predicate {hfc(Parent, Head, Except, Parent2)} to verify compliance with the HFC. This predicate is true only if the head features on the category Parent match those on its head Head, as required by the HFC. Some head features on Parent may be instantiated by this requirement; Parent2 is the resulting more restricted output feature structure.

The argument Except contains a list of head features for which the matching requirement is over-ridden. For example, in a rule such as that in Figure 5.2, the bar level of the head is different from that of the parent, as per GPSG theory (Sells, 1985, p. 86).

The GPSG head features known to IDEO's parser are AGR, BAR, N, PFORM, SUBCAT, SUBJ, V.

### 5.1.4 Control Agreement Principle

The CAP is enforced by {cap(Controller, Controllee, Controllee2)}. The predicate is true only if the AGR features of the Controller and Controllee categories unify. Controllee2 is the output category for the controllee, usually differing from Controllee in that its AGR feature value is defined.

### 5.1.5 Foot Feature Principle

The FFP is ascertained to hold for the tree with root Parent whose list of daughters is Daughters by the predicate {ffp(Parent, Daughters, Except, Parent2)}. The foot features on Parent must be just the unification of the foot features on the daughter categories. Foot features specified in the ID rule to construct Parent are exempt from this consideration, as per GPSG theory (Sells, 1985, p. 109). Except is the list of the foot features specified in the ID rule (the empty list in the case of my grammar rule 30; no foot features are specified on any category in the ID rule 159).

As promised in Section 4.1, if a parent has more than one daughter, ffp allows it nondeterministically to be ⟨focus, −⟩ regardless of the value of its daughters' focus features.

## 5.2 Morphology and the lexicon

Certain grammar rules introduce words. These implement lexical ID rules of GPSG (not shown in the toy grammar of Figure 4.1).

```
node(121, H, H4) ⟶
        {category(H2, lexical-head)},
        {unify(H, H2, struct(H3))},
        [Word],
        {morph(Word, W)},
        unify(W, struct([form(Word) | H3]), H4).
```

Figure 5.3: A typical lexical grammar rule

Rule 121 shown in Figure 5.3 recognizes lexical heads; `lexical-head` is an abbreviation for the category {⟨BAR, 0⟩, ⟨SUBCAT⟩ }. If this category unifies with the starting category `H` for the grammar rule (i.e., a lexical head in the input is consistent with the top-down expectations of the invoking grammar rule), then the input word `Word` is looked up in the lexicon, via morphological processing.

The implemented morphology is simplistic. If `Word` is stressed, indicated by the input token being a functor `stress(Lex)`, then its category is instantiated ⟨focus, +⟩, ⟨focus-in, +⟩, and ⟨p-set, true⟩ (see Section 4.2). Otherwise, its `p-set` feature value is obtained by copying its `assert` feature.

The morphology being simple, all inflected forms of a word require entries in the lexicon. Lexical entries pair words with categories that the lexical grammar rules use. The following features are present on these categories (when defined for the word in question): AGR, BAR, N, SUBCAT, V, label (see Section 5.3), assert, PFORM. For example, AGR, label, and assert are defined on determiner lexical entries, but N, SUBCAT, V, PFORM, and BAR are not. Focusing subjuncts are introduced nearly syncategorematically, their categories in the lexicon being defined only for the special features `fs`, `range-left` and `range-right` (see Section 4.1).

## 5.3 The semantic component

### 5.3.1 Data structures

Before proceeding to the semantic rules, it is necessary to describe IDEO's internal implementation of the frame representation described in Section 3.2. I employ a Horn-clause internal representation of frames. The Frail frame language used by Absity is implemented in a similar manner. Frames are translated into clauses in a first-order language. For example, the frame expressed (in Absity's notation) as

```
(160) (wash ?u
          (agent=(Ross)))
```

is stored in IDEO as the following conjunction of Prolog terms:

```
(161) frame(U,wash)
          & slot(U,agent,ross)
```

Shared variables tie together the different terms that are part of the representation of a single frame description. Thus the Prolog variables play an additional role in IDEO that the variables (?u, etc.) in Absity's frame notation do not.

IDEO represents frame descriptions as conjunctions of `frame` and `slot` functors, while frame statements are functors taking a frame description as one of their arguments:

(162) `framedet(`$\alpha$`, V, frame(V,`$\beta$`) & slot(V,`$\gamma$`,`$\delta$`) & ...)`

`V` is a shared variable tying this frame statement together, $\alpha$ is a frame determiner such as `the`, $\beta$ is a frame name, $\gamma$ is a slot name, and $\delta$ is an atom or a variable identifying another frame. This last case handles slot-fillers that are frame statements rather than atomic instances.

Two advantages accrue from using this internal frame representation instead of nested Prolog functors. The first is ease of construction. For example, a slot–filler pair can be added by simply conjoining a term to an existing frame description, rather than having to take apart (open) the frame description, insert the slot–filler pair, and re-assemble it. The second advantage is that I am able to express frames whose name is not known. This is necessary, for example, when a focusing subjunct focuses the verb and the sentence's p-set is a frame instance with an unspecified frame name. In effect, I have the capability to represent, as required, a quantification over frame names: to match or retrieve frames without specifying a frame name.

### 5.3.2 Semantic rules

As mentioned in Section 5.1, a semantic rule is invoked to compute the semantic objects for each node in parallel with the (syntactic) grammar rules. There is one semantic rule corresponding to each non-lexical grammar rule. Lexical grammar rules obtain values of semantic features from the information present in the lexicon.

Each semantic rule is responsible for computing the values of a node's three semantic features `assert`, `presupp`, and `p-set`, and the feature `label`, given the value of these features and the `fs` feature on the node's daughters. The `label` feature is a hook or identification tag on a semantic object. It is unified with the Prolog variable used to tie together the terms of the `assert` object of a node.

In most cases, the internal representation outlined in Section 5.3.1 allows the composition of the semantic objects of constituents simply by conjoining them and unifying their `label` feature values. For example, the `assert` feature value of the category for the VP

(163) [$_{VP}$ washed obj the dog]

is computed by conjoining the `assert` value `frame(U, washed)` of *washed* and that of *obj the dog*, `slot(V, patient, dog)`, and unifying the respective `label` values `U` and `V`. The resulting semantic object is

(164) `frame(U, washed) & slot(U, patient, dog).`

The `p-set` feature value of (163) is calculated similarly from the values of the p-set features on *washed* and *obj the dog*. When a category is ⟨focus, +⟩, however, its `p-set` feature value is computed as `true`. The reason for this is given in Section 5.4.

The **presupp** feature value is computed only for sentential categories (i.e., clauses), more specifically those containing a focusing subjunct (⟨fs, $\alpha$⟩ categories, where $\alpha \neq$ "–"). The computation of the semantic features is in general a bit more complicated for sentences containing focusing subjuncts. During these computations, a focusing subjunct operator will be applied. The features are computed just as set out in Section 4.2.

The extra types of sentential semantic objects introduced in Chapter 4 to handle focusing subjuncts obtain the following internal representation: A frame implication if P then A is represented using the specially defined Prolog operator `then`:

```
P then A
```

And (anew $?y \neq ?x$ (frame$\alpha$ $?y$)) is stored in IDEO as

```
framedet(anew, Y, X, frameα))
```

where `X` is expected to identify a frame in the knowledge base (see Section 5.4 below).

## 5.4 Knowledge-base semantics

My implementation accomplishes no useful purpose if it just translates strings of English to a frame representation without its own semantics. Hirst (1987) gives Absity's frame representation a semantics by defining the effects of passing a frame statement for processing to the underlying knowledge base. I do the same for IDEO's frame representation. I define the knowledge-base semantics of a frame statement as that defined by Hirst (see Section 4.3 above). That is, a frame statement (the $?y$ FrameDescription) retrieves the unique frame matching FrameDescription in the database. The frame determiner a asserts the existence of a new frame in the knowledge base matching its argument frame description, except that if such frames already exist, one is retrieved (arbitrarily).[2] The frame determiner `question` is treated as a request to retrieve from the knowledge base a frame matching its argument. I also implement the semantics described in Section 4.3 for frame implication and the frame determiner anew, the two new representation types necessitated by *only* and *even*.

### 5.4.1 Implementing frame determiners

Some explanation is required of the mechanisms to implement frame determiners in IDEO. Its knowledge base is in a different format from that underlying Absity. Just as its semantic

[2] This is not exactly Hirst's specification. Hirst's frame determiner the returns a new instance that is annotated as being one of a list of matching instances in the knowledge base. However, this annotation is not implemented in Frail, the frame representation language he uses (Hirst, 1987, p. 18).

objects are Horn clauses standing for frames, IDEO's knowledge base consists of Prolog clauses representing frame information.

IDEO's knowledge base consists of a collection of Prolog clauses of the form `frame(`$x$`, `$\alpha$`)` and `slot(`$x$`, `$\beta$`, `$z$`)`. $\alpha$ is a frame name and $\beta$ is a slot name; $x$ is a unique **frame label** atom identifying a frame. $z$ is a frame instance: either another frame label or an atom. A frame in the knowledge base is a set of clauses sharing the same frame label.

A frame description

(165) `frame(X, `$\alpha$`) & slot(X, `$\beta$`, `$z$`) & ...`

is **in the knowledge base** if each of its terms is a clause in the knowledge base, with like variables in the frame description unified. Retrieval from the knowledge base is therefore a matter of a Prolog proof of a frame description (viewed as a conjunction of clauses). Handing the knowledge base the frame statement

(166) `framedet(the, X, frame(X, `$\alpha$`) & slot(X, `$\beta$`, `$z$`) & ...)`

invokes the frame determiner function `the`, which succeeds if exactly one frame matching (165) is in the knowledge base.

Processing the frame statement

(167) `framedet(a, X, frame(X, `$\alpha$`) & slot(X, `$\beta$`, `$z$`))`

entails asserting the terms of (165) as Prolog clauses, that is, adding the frame description to the knowledge base, if it is not already in the knowledge base. Note that while a semantic object is a frame representation with *variables* linking its terms, when a frame description is inserted in the knowledge base these variable are replaced by *frame labels*. The successful knowledge base processing of a frame statement in fact results in the variables in the frame statement being instantiated to the appropriate frame label atoms.

This mechanism turns out to be essential to the implementation of the knowledge base semantics of the new semantics types introduced in Section 4.2.2: the frame implication and the frame determiner (anew $?y \neq ?x$) (Section 5.4.2).

### 5.4.2 Implementing the new semantic types

An *even* sentence that has its presupposition computed as

(168) `framedet(anew, Y, X, FrameDescription)`

has as its assertion

(169) `framedet(a, X, FrameDescription2)`

with `X` in (168) and (169) being the same variable (unified), and where the terms of `FrameDescription` are a subset of those of `FrameDescription2` (see "The sentential operators" in Section 4.2.2). Processing (169) against the knowledge base causes `X` to be instantiated to a frame label atom, $x$, identifying a frame matching `FrameDescription2`.

The processing of (168) is then meaningfully defined to be the assertion of the existence of a frame matching `FrameDescription`, but having a frame label other than $x$. That is, it is a distinct frame. Note that because it is generated by the *even* operator, it is possible for `FrameDescription` to be without a `frame` term, yet IDEO treats it as a valid frame description.

The knowledge base semantics for frame implications is a bit more complicated. Rooth's IL representation for the meaning of an *only* sentence used the implication operator. It should come as no surprise the processing of frame implications in IDEO's knowledge base involves the storage of Prolog *rule* clauses.

The knowledge base processes a frame implication `A then P` (where `A` and `P` are frame descriptions, that is, conjunctions of terms) by first verifying that no frame in the database that does not match `P` matches `A`. A frame matching `P` is retrieved from the knowledge base (see below) using the same operation as the frame determiner `a`. Then, a Prolog rule is inserted into the knowledge base to the effect that the frame description `P` matches all frames that `A` matches. This rule then becomes a constraint imposed on the knowledge base. Subsequent insertions in the knowledge base must be verified for consistency with this constraint before being completed. Thus the constraint effects a kind of "negative information" that can prohibit the insertion of certain (positive) instances into the knowledge base.

I mentioned that a frame matching `P` is *retrieved* from the knowledge base when processing the frame implication `A then P`. It is reasonable to hypothesize that any sentence producing such a semantic object, such as an *only* sentence, also produces the frame description `P` as a semantic object. Certainly the *only* operator conforms to this hypothesis: it produces sentences having the former object as the assertion and the latter as the presupposition. So there will always be a `P` to retrieve.[3]

The inference required to verify consistency with the knowledge base constraint imposed by a frame implication is tricky to implement. Suppose for example that the sentence (170.1) is interpreted, and the assertion expression (170.2) is obtained.

(170) 1. Only ROSS washed the dog.

```
2. if (frame(X, wash) & slot(X, patient, dog))
   then (frame(X, wash) & slot(X, patient, dog)
     & slot(X, agent, ross))
```

As a result, IDEO adds to its knowledge base a rule $R$ to the effect that if the frame description `(frame(X, wash) & slot(X, patient, dog))` matches a frame, then it is perforce the frame `wash1`, which is matched by the frame description

[3] The semantic interpreter should pass the presupposition to the knowledge base before the assertion.

(171) `(frame(X, wash) & slot(X, patient, dog) & slot(X, agent, ross)).`

Suppose IDEO subsequently interprets the sentence (172.1), arriving at the assert object (172.2).

(172) 1. Nadia washed the dog at home.

```
      2. framedet(a, X, frame(X, wash) & slot(X, patient, dog)
             & slot(X, agent, Nadia) & slot(X, location, home))
```

Clearly the insertion of this frame in the the knowledge base would violate the constraint imposed by the rule $R$ resulting from (170). Proving this is not easy. First, IDEO must check that the Horn clause `(frame(X, wash) & slot(X, patient, dog))` in the antecedent of the rule $R$ can be matched in (172.2), *not* vice versa, as would be the case when performing the usual retrieval operation using the frame determiner function `a`. Secondly, since this match succeeds, IDEO must verify that the frame description in (172.2) could indeed specify the frame instance `wash1`. Since it could not—and to detect this IDEO must know the constraint that a given frame can only have one filler per slot—the function call (172.2) fails on account of its failure to meet the constraint $R$. The nonmonotonic reasoning that is needed to check the consistency of the knowledge base against constraints is an active area of Artificial Intelligence research; see Ginsberg (1987).

A final observation is that the frame implications representing the semantics of an *only* sentence closely resembles McCord's (1982) only focalizer. The focalizer "means" that in all "cases" where its base argument holds, its focus argument holds. The particular frame implication computed for an *only* sentence "means" that if the sentence's p-set (a frame description) holds (of a frame in the knowledge base), then its assertion holds. As was shown in Section 4.2, McCord's arguments to the focalizer convey equivalent information to a sentence's p-set and its assertional content (logical form per Rooth).

## 5.5 Limitations

The program IDEO falls short of being a full implementation of the semantics described in chapter 4 in several ways. As well, it is somewhat less than ideal as an illustration of these semantics for a number of reasons.

- IDEO's grammar is limited. A particular serious omission is that it does not cover embedded sentences, whether they be relative clauses or sentence subjects.

- Because its grammar is limited, IDEO does not verify my semantics of focusing subjuncts in complex sentences. It does not test the semantics in cases in which either a focusing subjunct or its focus is contained within an embedded sentence.

- The program simply translates input sentences into sentential forms of a frame representation, and passes these on for processing against a knowledge base. No front end is implemented to make use of the semantic interpretation and knowledge base processing performed by Ideo. So, for instance, English answers are not generated for queries, or to indicate presuppositional failure.

- As implemented, Ideo does not process the presupp and assert semantic objects of a sentence differently.

- The grammar does not support negation. There is no provision for storing negative instances in the knowledge base.

- Inference to maintain knowledge base consistency by enforcing the constraints imposed by frame implications is not in place. A rough simplification is implemented instead.

## 5.6 Examples

A sample session with Ideo is included below. In this trace, output generated by the program or typed by the user is shown in typewriter font. As the example shows, Ideo accepts a sentence typed at the keyboard and outputs four different things. Firstly, it echoes the sentence in its internal Prolog format. Secondly, the GPSG category obtained for the sentence, which incorporates a parse tree for the sentence, is displayed. For the sake of readability, Ideo prints a simplified version of the category, which is actually a much larger structure than shown in the sample session below. Then, it prints separately the semantic representation of the sentence, which consists of several of the feature–value pairs of the sentence category. Lastly, it displays a message about the success or failure of each knowledge base function call resulting from processing the semantic representation of the sentence. In a question-and-answer system, these messages would be replaced by generated English answers.

The first sample sentence has no focusing subjuncts.

```
=============================================================
Welcome to Ideo.
=============================================================
Please enter a sentence, followed by a
carriage return, after the "|:" prompt.
Accented words are indicated by preceding them with a colon.
Enter ^D to leave Ideo.
-------------------------------------------------------------
|: Ross washed a dog.
-------------------------------------------------------------
>>> Saw the sentence:
  [ross, washed, a, dog, period]
-------------------------------------------------------------
>>> The category for this sentence is:
```

```
struct([
   subject(struct([
      head(struct([
         form(ross)
       ]))
   ])),
   vp(struct([
      head(struct([
         form(washed)
      ])),
      pp(struct([
         head(struct([
            form(obj),
         ])),
         np(struct([
            det(struct([
               form(a),
            ])),
            head(struct([
               form(dog)
            ]))
         ]))
      ]))
   ]))
])
```

Only the parse-tree information carried on the category is displayed. The structure shown above corresponds to the parse tree in Figure 5.4.

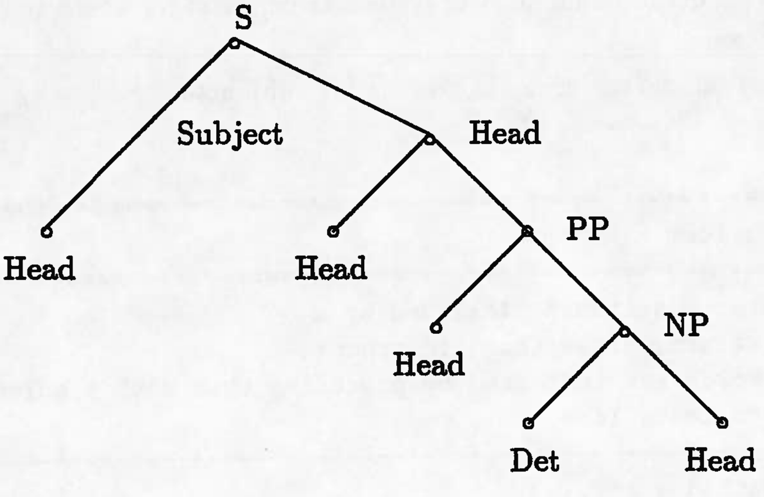

Figure 5.4: Parse tree for *Ross washed a dog.*

```
>>> The semantic representation for this sentence is:
  assert(framedet(a, X,
    frame(X, wash) & slot(X, agent, ross) & slot(X, patient, Z)
    & framedet(a, Y, frame(Y, dog))))
  presupp(true)
  p-set(framedet(a, X,
    frame(X, wash) & slot(X, agent, ross) & slot(X, patient, Y)
    & framedet(a, Y, frame(Y, dog))))
  fs(-)
>>> OK? yes
```

The sentence "asserts" that there is a washing having Ross as its agent and a dog as its patient. It has no "presuppositions", since IDEO only knows about the non-asserted meanings of focusing subjuncts. There is no focusing subjunct in the sentence under consideration, as the feature–value pair fs(-) indicates. The p-set of the sentence is the same as its assertion, since it contains no intonational accent.

The prompt OK? is a request for approval from the user for the parse tree and semantic representation IDEO that printed out. If the user does not enter yes, the sentence is re-parsed. If another parse of the sentence exists, the new parse tree and semantic representation are printed out.

```
---------------------------------------------------------------
>>> Inserted frame "dog1"
  frame(dog1, dog)
>>> Inserted frame "wash1"
  frame(wash1, wash)
  slot(wash1, agent, ross)
  slot(wash1, patient, dog1)
```

IDEO now has in its knowledge base a dog frame dog1 and a washing frame wash1 whose agent is Ross and whose patient is dog1.

The next sample sentence contains the focusing subjunct *only*.

```
===============================================================
Please enter a sentence, followed by a
carriage return, after the "|:" prompt.
Accented words are indicated by preceding them with a colon.
Enter ^D to leave Ideo.
---------------------------------------------------------------
|: Only :Ross washed the dog.
```

The colon preceding the word Ross tells IDEO that the word is intonationally stressed.

```
------------------------------------------------------------
>>> Saw the sentence:
  [only, stress(ross), washed, the, dog, period]
------------------------------------------------------------
>>> The category for this sentence is:
  struct([
     fs(only),
     subject(struct([
        adjunct(struct([
           fs(only), form(only)
        ])),
        head(struct([
           head(struct([
              focus(+),
              form(stress(ross))
            ]))
        ]))
     ])),
     vp(struct([
        head(struct([
           form(washed)
        ])),
        pp(struct([
           head(struct([
              form(obj),
           ])),
           np(struct([
              det(struct([
                 form(the),
              ])),
              head(struct([
                 form(dog)
              ]))
           ]))
        ]))
     ]))
  ])
------------------------------------------------------------
>>> The semantic representation for this sentence is:
  assert(
    if frame(X, wash) & slot(X, patient, Y)
      & framedet(the, Y, frame(Y, dog))
    then frame(X, wash) slot(X, agent, ross) & slot(X, patient, Y)
      & framedet(the, Y, frame(Y, dog)))
  presupp(framedet(a, X,
    frame(X, wash) & slot(X, agent, ross) & slot(X, patient, Y)
    & framedet(the, Y, frame(Y, dog))))
  p-set(framedet(a, X,
    frame(X, wash) & slot(X, patient, Y)
    & framedet(the, Y, frame(Y, dog))))
  fs(only)
>>> OK? yes
------------------------------------------------------------
>>> Retrieved frame "dog1"
  frame(dog1, dog)
>>> Found frame "wash1"
  frame(wash1, wash)
  slot(wash1, agent, ross)
  slot(wash1, patient, dog1)
>>> Inserted rule "rule1"
  if
    frame(X, wash) & slot(X, patient, dog1)
  then
    X = wash1
```

The knowledge base now is constrained by the rule `rule1`. This says that if a frame `X` satisfies the frame description `frame(X, wash) & slot(X, patient, dog1)` then it must be the frame `wash1` (whose agent slot is filled by `ross`).

The next sentence contains information that would violate the constraint imposed by `rule1`.

```
============================================================
Please enter a sentence, followed by a
carriage return, after the "|:" prompt.
Accented words are indicated by preceding them with a colon.
Enter ^D to leave Ideo.
------------------------------------------------------------
|: Nadia washed the dog.
------------------------------------------------------------
>>> Saw the sentence:
  [nadia, washed, the, dog, period]
------------------------------------------------------------
>>> The category for this sentence is:
  struct([
    subject(struct([
      head(struct([
        form(nadia)
       ]))
    ])),
    vp(struct([
      head(struct([
        form(washed)
      ])),
      pp(struct([
           head(struct([
              form(obj),
           ])),
           np(struct([
              det(struct([
                 form(the),
              ])),
              head(struct([
                 form(dog)
              ]))
           ]))
        ]))
     ]))
  ])
-------------------------------------------------------------
>>> The semantic representation for this sentence is:
  assert(framedet(a, X,
    frame(X, wash) & slot(X, agent, nadia) & slot(X, patient, Z)
    & framedet(the, Y, frame(Y, dog))))
  presupp(true)
  p-set(framedet(a, X,
    frame(X, wash) & slot(X, agent, ross) & slot(X, patient, Y)
    & framedet(the, Y, frame(Y, dog))))
  fs(-)
>>> OK? yes
-------------------------------------------------------------
>>> Retrieved frame "dog1"
  frame(dog1, dog)
>>> Assert failure: could not insert frame
  frame(X, wash)
  slot(X, agent, nadia)
  slot(X, patient, dog1)
  due to conflict with rule "rule1"
```

IDEO is unable to insert into the knowledge base a frame about Nadia washing the dog, because this would violate the constraint imposed by the rule rule1. That is, since *Only* ROSS *washed the dog*, it is not possible that *Nadia washed the dog.*

The next sentence, which contains the focusing subjunct *even*, also violates the constraint imposed by rule1.

```
=============================================================
Please enter a sentence, followed by a
carriage return, after the "|:" prompt.
Accented words are indicated by preceding them with a colon.
Enter ^D to leave Ideo.
-------------------------------------------------------------
```

```
|: :Ross even washed the dog.
------------------------------------------------------------
>>> Saw the sentence:
  [stress(ross), even, washed, the, dog, period]
------------------------------------------------------------
>>> The category for this sentence is:
  struct([
     fs(even),
     subject(struct([
        adjunct(struct([
           fs(even), form(even)
        ])),
        head(struct([
           head(struct([
              focus(+),
              form(stress(ross))
            ]))
        ]))
     ])),
     vp(struct([
        head(struct([
           form(washed)
        ])),
        pp(struct([
           head(struct([
              form(obj),
           ])),
           np(struct([
              det(struct([
                 form(the),
              ])),
              head(struct([
                 form(dog)
              ]))
           ]))
        ]))
     ]))
 ])
------------------------------------------------------------
>>> The semantic representation for this sentence is:
  assert(framedet(a, X,
    frame(X, wash) & slot(X, agent, ross) & slot(X, patient, Z)
    & framedet(the, Z, frame(Z, dog))))
  presupp(framedet(anew, Y, X,
    frame(Y, wash) & slot(Y, patient, Z)
    & framedet(the, Z, frame(Y, dog))))
  p-set(framedet(a, X,
    frame(X, wash) & slot(X, patient, Z)
    & framedet(the, Z, frame(Z, dog))))
  fs(only)
>>> OK? yes
-------------------------------------------------------------
>>> Retrieved frame "dog1"
  frame(dog1, dog)
>>> Found frame "wash1"
  frame(wash1, wash)
  slot(wash1, agent, ross)
  slot(wash1, patient, dog1)
>>> Presupposition failure: could not insert new frame
  frame(Y, wash)
  slot(Y, patient, dog1)
  due to conflict with rule "rule1"
```

The sentence *Even* Ross *washed the dog* "presupposes" that someone besides Ross washed the dog. However, Ideo has been told that *Only* Ross *washed the dog.* This "presuppositional failure" shows up when Ideo is unable to insert into the knowledge base a frame about washing the dog that is distinct from the from wash1 in which Ross is the agent. Doing so would violate the constraint imposed by the rule rule1.

The next sentence illustrates the importance of stress in determining the meaning of sentences containing focusing subjuncts. The sentence has the same word order as the previous sentence, but a different word is accented. The result is that this sentence, unlike the previous one, can be processed without violating the rule rule1.

```
=============================================================
Please enter a sentence, followed by a
carriage return, after the "|:" prompt.
Accented words are indicated by preceding them with a colon.
Enter ^D to leave Ideo.
-------------------------------------------------------------
|: Ross even washed the :dog.
-------------------------------------------------------------
>>> Saw the sentence:
  [ross, even, washed, the, stress(dog), period]
-------------------------------------------------------------
>>> The category for this sentence is:
  struct([
    fs(even),
    subject(struct([
      head(struct([
        form(ross)
      ]))
    ])),
    vp(struct([
        fs(even),
        adjunct(struct([
           fs(even), form(even)
        ])),
        head(struct([
           head(struct([
              form(washed)
           ])),
           pp(struct([
              head(struct([
                 form(obj),
              ])),
              np(struct([
                 focus(+),
                 det(struct([
                    form(the),
                 ])),
                 head(struct([
                    focus(+),
                    form(stress(dog))
                 ]))
              ]))
           ]))
        ]))
     ]))
  ])
------------------------------------------------------------
>>> The semantic representation for this sentence is:
  assert(framedet(a, X,
    frame(X, wash) & slot(X, agent, ross) & slot(X, patient, Z)
    & framedet(the, Z, frame(Z, dog))))
  presupp(framedet(anew, Y, X,
    frame(Y, wash) & slot(Y, agent, ross)))
  p-set(framedet(a, X,
    frame(X, wash) & slot(X, agent, ross)))
  fs(only)
>>> OK? no
```

This semantic representation corresponds to the reading in which the speaker means to presuppose that Ross washed other things besides the dog. The user replies no to the prompt OK? to indicate that this is not the reading he intends.

```
------------------------------------------------------------
>>> The category for this sentence is:
  struct([
     fs(even),
     subject(struct([
        head(struct([
           form(ross)
         ]))
     ])),
     vp(struct([
        fs(even),
        adjunct(struct([
           fs(even), form(even)
        ])),
        head(struct([
           focus(+),
           head(struct([
              form(washed)
           ])),
           pp(struct([
              focus(+),
              head(struct([
                 form(obj),
              ])),
              np(struct([
                 focus(+),
                 det(struct([
                    form(the),
                 ])),
                 head(struct([
                    focus(+),
                    form(stress(dog))
                 ]))
              ]))
           ]))
        ]))
     ]))
  ])
------------------------------------------------------------
>>> The semantic representation for this sentence is:
  assert(framedet(a, X,
    frame(X, wash) & slot(X, agent, ross) & slot(X, patient, Z)
    & framedet(the, Z, frame(Z, dog))))
  presupp(framedet(anew, Y, X,
    slot(Y, agent, ross)))
  p-set(framedet(a, X,
    slot(X, agent, ross)))
  fs(only)
>>> OK? yes
```

The user approves this semantic representation, which corresponds to the reading in which the speaker presupposes that Ross did other things besides washing the dog. Notice that the feature focus(+) percolates further up the parse tree in this parse than in the previous parse.

```
--------------------------------------------------------------
>>> Retrieved frame "dog1"
  frame(dog1, dog)
>>> Found frame "wash1"
  frame(wash1, wash)
  slot(wash1, agent, ross)
  slot(wash1, patient, dog1)
>>> Inserted new frame "f2"
  slot(wash2, agent, ross)
```

The sentence *Even Ross ⟨washed the* DOG⟩ "presupposes" that Ross did something besides washing the dog. Unlike the previous sentence, this sentence does not conflict with the earlier sentence *Only* ROSS *washed the dog.* IDEO is able to process it by inserted a new frame `f2` having Ross as its agent, but having an unspecified frame name and patient.

# Chapter 6

# Conclusions

## 6.1 Goals

In Section 1.4, I promised a semantics for focusing subjuncts that is compositional, computationally practical, able to differentiate between asserted and non-asserted meaning, sensitive to intonation, and cross-categorial. In this section I look at how well each of these objectives were met.

Firstly, the semantics I have outlined is compositional because it satisfies the characteristic conditions of compositionality.

- Each word and well-formed syntactic phrase is represented by some semantic object.
- The semantic representation of a syntactic phrase is a systematic function of the representation of its constituent words and/or phrases.

The semantic objects I employ happen to be composed of several parts. Because the construction of the semantic representation of sentences containing focusing subjuncts is accomplished by means of a special condition, the semantics is marginally less compositional than Absity's.

Secondly, the conversion of Rooth's semantics to employ a frame representation makes the semantics computationally practical. The implementation of the semantics in the program IDEO provides testimony to their computational practicality. Thirdly, the semantics is able to differentiate between asserted and non-asserted meaning, because it constructs a separate presupposition and assertion expression for each sentence.

The fourth objective is satisfied, as the determination of the focus of a focusing subjunct, a choice that crucially affects the meaning of the sentence containing it, depends on the distribution of the focus feature in the sentence. Since focus is recognized phonetically by intonational stress on words, the semantics is intonation-sensitive.

Lastly, the semantics employs a single lexical entry for each focusing subjunct, and a single sentential operator for each of *even* and *only*, to handle focusing subjuncts in a wide variety of constructions. My semantics handles focusing subjuncts in a greater number of constructions than any previous formal semantics.

## 6.2 Contributions to the field

The present work constitutes a step along the road to a better understanding of focusing subjuncts, and to a lesser extent, of the phenomenon of linguistic focus. The work of Rooth (1985) is extended, set in a formal syntactic framework, and adapted to a more tractable semantic framework. Ideas of McCord, Karttunen and Peters, and Jackendoff and of the author's device extend Rooth's theory.

This thesis also contributes to the understanding of compositional semantics. The rendering of Rooth's higher-order semantic forms in a frame representation is of interest. The semantics of focusing subjuncts involves other interesting knowledge representation problems that were raised in this thesis.

The semantics proposed in this thesis incorporates the following improvements over previous accounts of focusing subjuncts:

- Wider coverage of focusing subjuncts is accomplished than in any previous theory. *Even* and *only* are accounted for when they are adjoined to any maximal projection, not just NP and VP. There are valid cases of all such constructions in English.

- Ambiguities of scope arising from different distributions of the focus feature are detected.

- The presuppositions (or conventional implicature) and assertion introduced by a focusing subjunct are distinguished. Rooth conjoins them to obtain the logical form for a sentence. But I have argued that retaining the distinction provides a consumer of the output of the semantic interpretation, such as a pragmatic processor or a question-answering system, with useful information, and permits the correct handling of inheritance of the non-asserted meaning of embedded sentences.

- The meaning of focusing subjuncts is captured in a computationally practical representation. Rooth's semantics is adapted for a frame representation, instead of an intractable higher-order Intensional Logic. Second-order constructs such as $\lambda$-expressions and properties are avoided, replaced by under-specified semantic objects (frames) and latent sentential operators.

- By making focusing subjuncts sentential operators, a simpler semantic representation is possible without resorting, as McCord (1982) does, to a proliferation of operators able to modify the semantic objects of any and all phrases.

## 6.3 What was learned

This thesis arrives at certain conclusions about the semantics of focusing subjuncts. It employs a number of techniques that are useful in the semantic interpretation of focusing subjuncts as well as in computational semantic interpretation in general.

- Focus and stress information can be used to advantage in a semantic interpreter.

- Wider coverage than Rooth's is made possible by the syncategorematic introduction of focusing subjuncts and the relaxation of strong typing with respect to their arguments (adjuncts). Focusing subjuncts should not be treated syntactically as adverbs, because they have radically different semantics and different syntactic distribution.

- The hypothesis that focus may be optionally percolated to a parent node from a daughter explains the scope ambiguities observed in the interpretation of focusing subjuncts.

- It is my contention that the focus feature exhibits different behavior from that attributed to it by Rooth. Within the GPSG framework, focus is best (but not quite satisfactorily) explained as a head feature.

- Rooth's method of obtaining the translation of a focusing subjunct by using p-sets to select "domains of quantification" can be adapted to translating a sentence into a frame representation. The translations obtained are roughly equivalent to both Rooth's and McCord's (see Section 4.2).

- Treating focusing subjuncts as operators on sentential semantic forms makes the aforementioned translation possible.

- Extensions to an Absity-style frame representation are required for the translation of focusing subjuncts. Semantically, focusing subjuncts are not just passive objects for composition.

- The semantics of focusing subjuncts involves negative information. Dealing with focusing subjuncts in embedded sentences often involves the representation of belief reports and of agents' beliefs and other attitudes. In the absence of a proper account of these phenomena, particularly within frame representation systems, the semantics of focusing subjuncts outlined in this paper is incomplete. I contend that focusing subjuncts introduce no knowledge representation problems of their own. However, their semantics involves negation, identity, and beliefs, all of which are problematic for KR.

## 6.4 Further Research

Many issues are left unresolved in my account of focusing subjuncts. Some simplifying assumptions were made and other issues were simply not dealt with. Further work in the analysis of focusing subjuncts needs to address the following points.

- Although I have expanded the syntactic phenomena that can be handled, the syntax of focusing subjuncts is even more complicated than assumed in the thesis (see Section 4.1). Conditions on the distribution of focusing subjuncts, especially within noun phrases and prepositional phrases, need to be formulated.

- An account is required of multiple focusing subjuncts within a clause.

- There is a richer intonational structure than just a stressed/unstressed dichotomy. Jackendoff (1972, p. 259), for instance, provides an account of multiple foci of focusing subjuncts that uses two different kinds of pitch accents.

- Realizations of the focus feature other than stress should be examined.

- There may be a relationship between the focus feature and information status (new vs. "given"). If there is, it may have consequences for the semantics of focusing subjuncts that need to be examined.

- A theoretical explanation of why differences in percolation of focus create ambiguity in certain places in the phrase structure of a grammar but not in others (e.g., PP from prepositional object) is needed.

- The status of non-asserted meanings of focusing subjuncts must be determined. If they are presuppositions, then they should be combined during interpretation with the other presuppositions of a sentence, such as existential presuppositions introduced by noun phrases.

- My semantics constructs separate semantic expressions to represent the asserted and non-asserted meanings of focusing subjuncts. However, I do not specify a different treatment of these two expressions in the knowledge base. With reservations, I make the simplification that both of them are processed in the same way.

- A correct, complete processing of the semantics of focusing subjuncts against a knowledge base depends on having an inference engine that can maintain the consistency of a constrained knowledge base. This is a problem for future research.

- I maintain that since in my semantics p-sets are not "properties", intensions are not a crucial part of the semantics of focusing subjuncts. However, while focusing subjuncts do not introduce intensions into meaning, other things do. If and when the knowledge representation problem of intensionality is solved, the relationship between focusing subjuncts and intensions should be re-examined.

- Similarly, when a satisfactory representation of time becomes available, the semantics of focusing subjuncts should be re-cast to take advantage of this representation.

- The pragmatics of focusing subjuncts should be looked at. The expectational meaning of focusing subjuncts like *even*, in particular, may involve pragmatics. P-sets may turn out to be a useful tool for pragmatics.

- The semantics of the focusing subjuncts other than *even* and *only* should be considered closely.

- Most of the words that act as focusing subjuncts are polysemous. This thesis does not contribute to the disambiguation of these words.

- An implementation of a much broader grammar of English than in this thesis would be useful to evaluate the semantics of focusing subjuncts. The limitations of the implementation written for this thesis are listed in Section 5.5.

## 6.5 Conclusion

This thesis has presented a semantics for focusing subjuncts that is compositional, computationally practical, able to differentiate between asserted and non-asserted meaning, sensitive to intonation, and cross-categorial. In so doing it has made progress towards the goal of a thorough understanding of focusing subjuncts. It further contributes a number of techniques that are useful in semantic interpretation in general.

# Appendix A

# Taxonomy of sentence adverbials

Adverbials are those sentence elements that are not the verb, subject, object, or complement. They can be realized syntactically by adverb phrases, prepositional phrases, and subordinate clauses. Adverbials contribute semantically to the sentence, being able to fill the following semantic roles (the adverbials are italicized in the example sentences) (Quirk et al., 1985, §8.2):

- Space: position, direction, distance.

  **(173)** They are strolling *in the park.*

- Time: position in time, duration, frequency, time relationship.

  **(174)** He had visited his mother *already* when I saw him yesterday.

- Process: manner, means, instrument, agent (in passive sentences).

  **(175)** The patient was *carefully* treated *by the nurse.*

  (two process adverbials, of manner and agent respectively)

- Respect: identifying a point of reference in respect of which the sentence derives its truth value.

  **(176)** He has always been frightened *of earwigs.*

- Contingency: cause, reason, purpose, result, condition, concession.

  **(177)** *Though he didn't read the book*, he acquired some knowledge of metaphysics.

- Modality: emphasis, approximation, restriction.

  **(178)** She has been enthusiastic *only* about her WORK.

- Degree: amplification, diminution, measure.

**(179)** She had worked *sufficiently* that day.

Quirk et al. (1985, §7.49, §8.121) divide adverbials into four categories, according to grammatical function.

**adjuncts** Adverbials whose (semantic) sentence role is of similar weight to other sentence elements (subject, object, etc.). Adjuncts are integrated within the clause.

**(180)** I will hand in my assignment *tomorrow.*

**subjuncts** Adverbials that are less independent, semantically and grammatically, than other sentence elements (and than adjuncts). Subjuncts are subordinate to some sentence element. Like adjuncts, they are integrated within the clause.

**(181)** *Even* Bob was there.

**disjuncts** Adverbials which play a semantically superordinate role within the sentence. They are syntactically more detached than other elements, and their scope extends over the whole sentence. Disjuncts are used to express an evaluation of what is said with respect form or meaning.

**(182)** *Of course*, nobody expected us to be here today.

**conjuncts** Conjuncts play a connective role. The speaker uses them to express his/her assessment of the relationship between two clauses.

**(183)** He worked hard, but he failed the exam *anyway.*

Since a subtype of subjuncts is the topic of interest, subjuncts need to be elaborated upon. The category is best defined by comparison with the category of adjuncts, which are also described in more detail in the following section.

## Adjuncts

Adjuncts can be identified by four characteristics that they share, but which are not usually found in other adverbials.

- They can be the focus of a cleft sentence or of a focusing subjunct (such as *only*).

**(184)** It was *because of his injury* that Hilda helped Tony.

**(185)** Hilda helped Tony only *because of his injury.*

- Adjuncts come within the scope of predication ellipsis (and of pro-forms).

**(186)** Fred *carefully* cleaned his teeth but Jonathan didn't.
= ··· but Jonathan didn't carefully clean his teeth.

- They can be elicited by question forms (including *wh*-questions).

  **(187)** Why did Hilda help Tony? *Because of his injury.*

- They can be contrasted in alternative interrogation or negation, as in the following example:

  **(188)** I won't see you *tomorrow*, but I'll see you *when I visit in December.*

### Subjuncts

Quirk et al. (1985, §8.88) divide subjuncts into six subcategories.

**viewpoint subjuncts** According to Quirk et al. (1985, §8.89), "Viewpoint subjuncts can be roughly paraphrased by 'if we consider what we are saying from an [adjective] point of view' or 'if we consider what we are saying from the point of view of [noun phrase]'." Viewpoint subjuncts are often realized by adverbs derived from adjectives by a *–ly* ending, or from nouns by a *–wise* suffix.

**(189)** *Architecturally*, it is a magnificent conception.

**(190)** *Weatherwise*, we are going to have a bad time this winter.

**courtesy subjuncts** These adverbials turn questions into requests, or tone down the abruptness of a command: *kindly, cordially, graciously, please.*

Subjuncts in these first two categories are called **wide orientation** subjuncts, because their (semantically) subordinate role applies to the whole clause in which they appear. The remaining categories of subjuncts are **narrow orientation.** They apply semantically only to an individual clause element (usually the subject).

**item subjuncts** are the subjuncts that are semantically subordinate to a specific constituent or clause element (usually the subject).

**(191)** *Carefully*, Leslie greeted the stranger in an offhand way.

In the above sentence, *carefully* is an item subjunct subordinate to the subject *Leslie*, if the sentence is taken to mean "Leslie was being careful as he greeted ...".

**emphasizers** are "subjuncts concerned with expressing the semantic role of *modality* which have a reinforcing effect on the truth value of the clause or part of the clause to which they apply" (Quirk et al., 1985, §8.99). They add force to the constituent they modify, which need not be gradable.

**(192)** He *really* may have injured innocent people.
= It is really possible that he has injured ⋯

**intensifiers** are "broadly concerned with the semantic category of *degree*" (Quirk et al., 1985, §8.104). Intensifiers modify gradable constituents, which distinguishes them from emphasizers. Their effect is to select some point on the intensity scale upon which the constituent is graded.

**(193)** They *greatly* admire his music.

**focusing subjuncts** draw attention to a part of a sentence. They are usually realized by adverbs, but occasionally by prepositional phrases. The item that is focused is usually 'new' information (Quirk et al., 1985, §8.116). They fill the semantic role of *modality*, having the effect of emphasizing or qualifying the truth value of a statement.

**(194)** I was *simply* taking my dog for a walk.

**(195)** The workers, *in particular*, are dissatisfied with the government.

Quirk et al. (1985, §8.116) divide focusing subjuncts into the following categories:

**restrictives** indicate that the utterance is true of the part focused. There are two subtypes:

**exclusives,** which restrict the utterance to apply *exclusively* to the part focused: *alone, exactly, exclusively, just, merely, only, precisely, purely, simply, solely.*

**particularizers,** which restrict the utterance to apply *predominantly* to the part focused: *chiefly, especially, largely, mainly, mostly, notably, particularly, primarily, principally, specifically, at least, in particular.*

**additives** indicate that the utterance is *additionally* true of the part focused: *again, also, either, equally, even, further, likewise, neither, nor, similarly, too, as well, in addition.*

## References

Anderson, S. R. (1972). How to get *even*. *Language*, 48:893–906.

Austin, John L. (1963). Performative-constative. In Caton, Charles Edwin, editor, *Philosophy and Ordinary Language*. University of Illinois Press, Urbana.

Bergmann, Merrie (1981). *Only*, *even*, and clefts in two-dimensional logic. In *Proceedings, The Eleventh International Symposium on Multiple-Valued Logic*, University of Oklahoma. pages 117–123. Proceedings also known as *IEEE 1981 Multiple Valued Logic*.

Charniak, Eugene (1981). A common representation for problem-solving and language-comprehension information. *Artificial Intelligence*, 16(3):225–255. Also published as technical report CS–59, Department of Computer Science, Brown University, July 1980.

Chomsky, Noam (1976). Conditions on rules of grammar. *Linguistic Analysis*, 2(4).

Chomsky, Noam (1981). *Lectures on Government and Binding*. Foris, Dordrecht.

Chomsky, Noam and Lasnik, Howard (1977). Filters and control. *Linguistic Inquiry*, 8(3).

Dik, Simon, Hoffmann, Maria E., de Jong, Jan R., Djiang, Sie Ing, Stroomer, Jarry, and de Vries, Laurens (1981). On the typology of focus phenomena. In Hoekstra, T., van der Hulst, H., and Moortgat, M., editors, *Perspectives on Functional Grammar*. Foris Publications, pages 41–74. [Book also published as *GLOT* 3 (1980) (Leiden Linguistic Journal)].

Dowty, David R., Wall, Robert E., and Peters, Stanley (1981). *Introduction to Montague Semantics*. D. Reidel Publishing Company.

Evans, Roger (1985). Program—a development tool for GPSG grammars. *Linguistics*, 23(2):213–243.

Fischer, Susan D. (1968). On cleft sentences and contrastive stress. Ms., MIT.

Fraser, Bruce (1971). An analysis of 'even' in English. In Fillmore, Charles J. and Langedoen, Terence D., editors, *Studies in Linguistic Semantics*. Holt, Rinehart and Winston Inc., pages 150–178.

Gazdar, Gerald, Klein, Ewan, Pullum, Geoffrey K., and Sag, Ivan (1985). *Generalized Phrase Structure Grammar*. Harvard University Press.

Ginsberg, Matthew L., editor (1987). *Readings in Nonmonotonic Reasoning*. Morgan Kaufmann Publishers, Inc.

Grice, H. Paul (1975). Logic and conversation. In Cole, Peter and Morgan, Jerry L., editors, *Speech Acts*, volume 3 of *Syntax and Semantics*. Academic Press, pages 41–58.

Gussenhoven, Carlos (1983). Focus, mode and the nucleus. *Journal of Linguistics*, 19:377–417.

Herzberger, Hans G. (1973). Dimensions of truth. *Journal of Philosophical Logic*, 2(4):535–556.

Hirschberg, Julia and Pierrehumbert, Janet (1986). The intonational structuring of discourse. In $24^{th}$ *Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference.* pages 136–143.

Hirst, Graeme (1981a). *Anaphora in Natural Language Understanding: A Survey*, volume 119 of *Lecture notes in computer science.* Springer-Verlag, New York. Also Technical report 79–2, Department of Computer Science, University of British Columbia, 1 May 1979. Also MSc thesis, Department of Engineering Physics, Australian National University, June 1979.

Hirst, Graeme (1981b). Discourse-oriented anaphora resolution in natural language understanding: A review. *American Journal of Computational Linguistics*, 7(2):85–98.

Hirst, Graeme (1987). *Semantic Interpretation and the Resolution of Ambiguity.* Cambridge University Press.

Horn, Laurence R. (1969). A presuppositional analysis of *only* and *even.* In Binnick, Robert I., Davison, Alice, Green, Georgia, and Morgan, Jerry, editors, *Papers from the Fifth Regional Meeting of the Chicago Linguistics Society.* Chicago Linguistic Society, pages 98–107.

Horton, Diane and Hirst, Graeme (1988). Presuppositions as beliefs. In *International Conference on Computational Linguistics, COLING 88, Proceedings*, Budapest. pages 255–260.

Jackendoff, Ray S. (1972). *Semantic Interpretation in Generative Grammar.* The MIT Press.

Karttunen, Lauri and Peters, Stanley (1979). Conventional implicature. In Oh, Choon-Kyu and Dinneen, David A., editors, *Presupposition*, volume 11 of *Syntax and Semantics.* Academic Press, pages 1–56.

Katz, Jerrold J. and Postal, Paul M. (1964). *An Integrated Theory of Linguistic Descriptions.* Cambridge.

Kempson, Ruth M. (1975). *Presupposition and the Delimitation of Semantics.* Cambridge University Press.

Ladd, Robert D. (1983). Even, focus and normal stress. *Journal of Semantics*, 2(2):157–170.

Levinson, Stephen C. (1983). *Pragmatics.* Cambridge University Press.

McCord, Michael C. (1982). Using slots and modifiers in logic grammars for natural language. *Artificial Intelligence*, 18:327–367. Originally Tech. Report, Computer Science Department, University of Kentucky, 1980.

McCord, Michael C. (1986). Focalizers, the scoping problem, and semantic interpretation rules in logic grammars. In van Caneghem, Michael and Warren, David H. D., editors, *Logic Programming and its Applications.* Ablex Publishing Corporation, pages 223–243.

Montague, Richard (1973). The proper treatment of quantification in ordinary English. In Hintikka, Kaarlo Jaakko Juhani, Moravcsik, Julius Matthew Emil, and Suppes, Patrick Colonel, editors, *Approaches to Natural Language: Proceedings of the 1970 Stanford workshop on grammar and semantics.* D. Reidel, pages 221–242. Also in Thomason, Richmond Hunt (ed.), *Formal philosophy: Selected papers of Richard Montague.* Yale University Press (1974): 247–270.

Pereira, Fernando C.N. and Shieber, Stuart M. (1987). *Prolog and Natural-Language Analysis.* Center for the Study of Language and Information.

Phillips, John D. and Thompson, Henry S. (1985). GPSGP—a parser for generalized phrase structure grammars. *Linguistics*, 23(2):245–261.

Pierrehumbert, Janet Breckenridge (1980). *The Phonology and Phonetics of English Intonation.* PhD thesis, Department of Linguistics and Philosophy, Massachusets Institute of Technology.

Quirk, Randolph, Greenbaum, Sidney, Leech, Geoffrey, and Svartvik, Jan (1985). *A Comprehensive Grammar of the English Language.* Longman.

Rooth, Mats Edward (1985). *Association with Focus.* PhD thesis, Department of Linguistics, University of Massachusets.

Selkirk, Elisabeth O. (1984). *Phonology and Syntax.* MIT Press.

Sells, Peter (1985). *Lectures on Contemporary Syntactic Theories.* Center for the Study of Language and Information.

Shieber, Stuart M. (1986). *An Introduction to Unification-Based Approaches to Grammar.* Center for the Study of Language and Information.

Sidner, Candace Lee (1979). *Towards a computational theory of definite anaphora comprehension in English discourse.* PhD thesis, Department of Electrical Engineering, Massachusetts Institute of Technology.

Stalnaker, Robert (1974). Pragmatic presuppositions. In Munitz, M. K. and Unger, P. K., editors, *Semantics and Philosophy.* New York University Press, New York, pages 197–214.

Walker, Adrian, editor, McCord, Michael, Sowa, John F., and Wilson, Walter G. (1987). *Knowledge Systems and Prolog: A Logical Approach to Expert Systems and Natural Language Processing.* Addison-Wesley Publishing Company, Inc.

Wilson, Deirdre and Sperber, Dan (1979). Ordered entailments: An alternative to presuppositional theories. In Oh, Choon-Kyu and Dinneen, David A., editors, *Presupposition*, volume 11 of *Syntax and Semantics.* Academic Press, pages 299–323.

Wong, Douglas (1981a). Language comprehension in a problem solver. In 7th *International Joint Conference on Artificial Intelligence, Proceedings*, Vancouver, British Columbia. pages 7–12.

Wong, Douglas (1981b). *On the Unification of Language Comprehension with Problem Solving.* PhD thesis, Department of Computer Science, Brown University. Available as technical report CS–78.

Woods, William Aaron (1967). *Semantics for a question-answering system.* PhD thesis, Harvard University. Reprinted as a volume in the series Outstanding dissertation in the Computer Sciences, Garland Publishing, New York, 1979.