Resolving Scope Ambiguity in Coordination with Minimalist Grammars

Irina Agafonova
Michigan State University

Abstract. One of the key problems in computational linguistics is resolving linguistic ambiguities of all kinds. This paper addresses a computational aspect of scope ambiguity in coordinate structures. I adopt a Minimalist Grammars formalism (Stabler, 1997; Stabler and Keenan, 2003; Kobele, 2006) to derive scope ambiguities and suggest a new explanation of scope effects in coordinate structures.

1 Introduction

A key task of computational linguistics is to resolve various kinds of ambiguities, e.g., lexical ambiguities, scope ambiguities, structural ambiguities, and attachment ambiguities. Ambiguities in natural language create possible readings which can grow exponentially. For example, the sentence in (1) has two scope elements, i.e. two quantifiers something and everyone. The sentence allows for 2! (two factorial) possible readings which equal to narrow (surface scope) and wide (inverse scope) scope readings of the object, paraphrased in (1a) and (1b).

(1) Something devoured everyone.
   a. There is something that devoured everyone.
   b. For each person, there is something that devoured him.

One way to deal with ambiguities in natural language when processed by a man or a machine is to enumerate all possible interpretations first and test their acceptability afterwards. However, the exponential growth of alternative readings makes such an approach inefficient and often infeasible.

Recently, other formalisms have been introduced to deal with ambiguities in natural language (e.g. Alshawi, 1990; Geurts and Rentier, 1993; Reyle, 1993; Bos, 1996; Muskens, 1999; Egg et al., 2001; Erk, 2002; Copestake et al., 2005). Common to all these formalisms is the use of underspecification techniques, which avoid the problem of exponential alternatives. The main idea underlying underspecification is to derive a single (constrained) description of all readings instead of generating all possible readings. In this paper, I address scope ambiguity in coordination. I adopt a Minimalist Grammars formalism (Stabler, 1997; Stabler and Keenan, 2003; Kobele, 2006), which uses underspecification in semantic representations. I use the formalism to account for different readings in gapping constructions with disjunction embedded under a modal verb.
The paper is organized as follows. Section 2 introduces a Minimalist Grammars formalism and direct compositionality. A scope ambiguity puzzle in coordination and a solution to the puzzle within the Minimalist Grammars formalism are presented in section 3 and 4, respectively. Section 5 concludes.

2 Interpretation and Minimalist Grammars

In this section, we introduce a grammar formalism – Minimalist Grammars (Stabler, 1997; Stabler and Keenan, 2003; Kobele, 2006). We adopt this grammar formalism to derive scope ambiguities in coordination and to formalize an alternative-based semantics approach to disjunction (Alonso-Ovalle, 2006; Hulsey, 2008). The new approach sheds light on a question whether alternative semantics (Hamblin, 1973) might be formalized.

2.1 Minimalist Grammars: A formal definition

Minimalist Grammars (MGs) is a grammar formalism that relies on a generative approach to language (Chomsky, 1965, 1995, 2004). An MG is a five-tuple $G = (\Sigma, F, Types, Lex, \mathcal{F})$, which consists of an alphabet, a set of features, categorial types, a lexicon and two generating functions – merge and move. (2) provides a formal definition of MGs (from Stabler and Keenan, 2003, p. 346).

(2) Definition. A Minimalist Grammar $G = (\Sigma, F, Types, Lex, \mathcal{F})$, where

- Alphabet $\Sigma \neq \emptyset$
- Features $F = \text{base}$ (basic features, $\neq \emptyset$)
  $\cup \{ f | f \in \text{base} \}$ (selection features)
  $\cup \{ + f | f \in \text{base} \}$ (licensor features)
  $\cup \{ - f | f \in \text{base} \}$ (licensee features)
- $Types = \{ ::, : \}$ (lexical, derived)
- Lexicon $Lex \subseteq C^+$ is a finite subset of $\Sigma^* \times \{::\} \times F^*$.
- Generating functions $\mathcal{F} = \{ \text{merge, move} \}$, partial functions from $E^*$ to $E$, where $E$ stands for Expressions.

MGs generate tuples of categorized strings or ‘chains’. A chain has a type and features such that $Chains C = \Sigma^* \times Types \times F^*$. A sequence of chains forms an expression such that $Expressions E = C^+$. Each expression is nonempty and finite. The deduction rules for the merge and move functions are given in (3) and (4), respectively (from Stabler and Keenan, 2003, p. 347).

(3) $\text{merge} : (E \times E) \rightarrow E$ is the union of the following three functions, for $s, t \in \Sigma^*$,

- $\cdot \in \{;::\}$, $f \in \text{base}$, $\gamma \in F^*$, $\delta \in F^+$, and chains $\alpha_1, \ldots, \alpha_k$, $\iota_1, \ldots, \iota_l (0 \leq k, l)$

a. $s := f \gamma \cdot f, \alpha_1, \ldots, \alpha_k$  
$st : \gamma, \alpha_1, \ldots, \alpha_k$  
$merge 1$  
(concatenation of the strings)
(4) move: \( E \rightarrow E \) is the union of the following two functions, for \( s, t \in \Sigma^* \), \( f \in \text{base} \), \( \gamma \in F^* \), \( \delta \in F^+ \), and chains \( \alpha_1, \ldots, \alpha_k, \iota_1, \ldots, \iota_l \) (0 \( \leq k, l \)); none of \( \alpha_1, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_k \) has \( -f \) as its first feature (the shortest move condition)

\begin{align*}
\text{a.} & \quad s := f_{\gamma, \alpha_1, \ldots, \alpha_k} t : f_{\iota_1, \ldots, \iota_l} \quad \text{move1} \\
\text{b.} & \quad s := f_{\gamma, \alpha_1, \ldots, \alpha_k} t : f_{\delta, \iota_1, \ldots, \iota_l} \quad \text{move2} \\
\text{c.} & \quad s := f_{\gamma, \alpha_1, \ldots, \alpha_k} t : f_{\iota_1, \ldots, \iota_l} \quad \text{merge2} \quad \text{(concatenation of the strings)} \\
\text{d.} & \quad s := f_{\gamma, \alpha_1, \ldots, \alpha_k} t : f_{\delta, \iota_1, \ldots, \iota_l} \quad \text{merge3} \quad \text{(empty string concatenation)}
\end{align*}

Language \( L(G) = \text{closure}(\text{Lex}, F) \). For any \( f \in F \), the strings of category \( f \), \( S_f(G) = \{ s | s \cdot f \in L(G) \text{ for some } \cdot \in \text{Types} \} \).

### 2.2 Minimalist Grammars: An example

In MGs expressions are build by the *merge* and *move* operations. The two operations are feature-driven. As an example, consider a transitive sentence (Kobele, 2006, p. 22) and a MG for this sentence (5b).

\( \text{(5)} \)

\[ \text{a. John devoured the ointment.} \]

\[ \text{b. john::d devoured::=d, =d, t} \]

\[ \text{the::=n, d ointment::n} \]

In (5), *john*, *devoured*, *the*, and *ointment* are lexical items, which we combine to build a complex expression or a sentence. Each lexical item has *categorial* \((f)\) and *selection* \((=f)\) features. The noun *ointment* has the categorial feature of a noun \( n \); *john* and *the* have the categorial feature of a determiner \( d \); the verb *devoured* has the categorial feature of being a tense phrase \( t \). In the example, only *the* and *devoured* have selection features. Selection features indicate that a lexical item requires another lexical item with a particular property. Whereas *the* selects for a noun, the verb *devoured* selects for two determiners. (6) is a derivation of the example sentence with the MG (5b), which shows how the selection features are checked in the derivation process.

\[ (\text{john,devoured, the ointment):t} \]

\[ ([], \text{devoured, the ointment}):=d \ t \quad \text{john::d} \]

\[ \text{devoured::=d =d t} \]

\[ ([], \text{the, ointment):d} \]

\[ \text{the::=n d ointment::n} \]
Now, let’s take a look at another example, which involves the move operation (4). (7a) is an intransitive sentence with a MG in (7b).

(7)  a. John arrived.
     b. john::d, -k arrive::=d, v ‘-ed’:::=v, +k, t

The following derivation steps are involved in building the sentence. First, we merge John and arrive as in (8). The label > indicates the head of the expression (the verb arrive) by pointing toward it.

(8) <
    <
    arrive:v john:-k

Then, we merge the derived expression (8) with a tense head (9).

(9) <
    <
    arrive -ed:t +k <
    john:-k

Notice, that both the tense head and John are marked for features that initiate movement. The tense head has the +k feature, which licenses movement (the licensor). The proper name John, on the other hand, has the -k feature (the licensee), which triggers movement of the DP as soon as the licensor feature is available. The next step in the derivation brings John into SpecTP position by movement operation and checks the -k feature on the subject DP as shown in (10).

(10)>
    john: <
    arrive -ed:t <

An X-bar tree view is provided in (11).

(11)

\[ \begin{array}{c}
\text{TP} \\
\mid \text{DP}(0) \mid \text{T'} \\
\mid \text{D'} \mid \text{T} \mid \text{vP} \\
\mid \text{D} \mid \text{v} \mid \text{T} \mid \text{v'} \\
\mid \text{John} \mid \text{arrive} \mid \text{-ed} \mid \text{v} \mid \text{DP} \\
\mid \text{T} \mid \text{t(0)} \\
\end{array} \]

\footnote{There is a head movement of arrive to the tense head of the TP. For more details regarding the head movement see Kobele (2006).}
2.3 Direct compositionality and MGs

A version of MGs proposed in Kobele allows for direct compositionality of derivation trees. According to this approach, items are interpreted as they move through the derivation, including their intermediate positions. The idea is implemented by associating a semantic value with each feature of an expression. Objects are interpreted as each feature is checked. As an example, consider the following sentence in (12a) and a MG for the sentence (Kobele, 2006, p. 75).

(12) a. George shaved some abbot.
    b. `george::d some::=n d -k -q abbot::n
       shaved::=d v c:::=v +k =d +q voice

The sentence has the derivation in (13), which shows checking of relevant features in the derivation process.

(13)

The following modes of semantic combination (14), associated with the merge and move operations, are used to provide direct compositional semantics of expressions (Kobele, 2006).

(14) a. \[\text{merge}(\alpha, \beta)\] \rightarrow [\alpha]( [\beta]) \quad (FA)
    b. \[\text{merge}(\alpha, \beta)\] \rightarrow [\beta]( [\alpha]) \quad (BA)
    c. \[\text{merge}(\alpha, \beta)\] \rightarrow [\alpha]( [\beta]) \quad \text{store}(\alpha)^\wedge \text{store}(\beta) \quad (FA)
    d. \[\text{merge}(\alpha, \beta)\] \rightarrow [\beta]( [\alpha]) \quad \text{store}(\alpha)^\wedge \text{store}(\beta) \quad (BA)
    e. \[\text{merge}(\alpha, \beta)\] \rightarrow [\alpha](x_i) \quad \text{store}(\alpha)^\wedge \text{G}( [\beta])(\lambda_i)^\wedge \text{store}(\beta) \quad (\text{Store})
    f. \[\text{move}(\alpha)\] \rightarrow [\alpha] \quad \text{store}(\alpha) \quad (\text{Id})
    g. \[\text{move}(\alpha)\] \rightarrow Q([\alpha]) \quad \text{store}(\alpha) – Q \quad (\text{Retrieve})

The derivation proceeds as follows. First, we merge some and abbot. The denotation of the noun applies (by function application (14a)) to the function denoted by the determiner, resulting in some(abbot). Next, we merge shave with some abbot. The denotation of shave (a function from individuals to predicates) cannot combine with the denotation of some abbot (a function from predicates to assignments). To tackle the problem Kobele (2006) suggests to use some storage

\(^2\) In (14g), \(Q\) is the stored meaning of the moving constituent.
mechanism (similar to Cooper, 1983) by feeding a variable to the denotation of the verb and storing the meaning of the DP for later interpretation (14e). The resultant denotation of the VP shave some abbot is shave(x₀) with the function G(some(abbot))(λ₀) in store. Then, we merge the voice head ϵ := v + k = d + q voice with the VP (14d). Now, some abbot moves to check its case feature -k (14f). The stored meaning is not retrieved at this point. The subject george is merged next (14d). The result of the subject merge is the set of assignments shave(x₀)(g), with stored G(some(abbot))(λ₀). Now, some abbot moves to check its -q feature (14g). We retrieve the stored meaning of the DP and apply the set of assignments shave(x₀)(g) to the stored G(some(abbot))(λ₀) as in (15).

\[ G(\text{some(abbot)})(\lambda_0)(\text{shave}(x_0)(g)) = \text{some(abbot)}(\lambda_0(\text{shave}(x_0)(g))) \]
\[ = \{ h: \text{for some } f \in [G \to E], g \text{ shaved } f(h) \text{ and } f(h) \text{ is an abbot} \} \]

Another relevant example involves quantifier scope interaction (Kobele, 2006, p. 80) and introduces the notion of the underspecified semantic representation in MGs. The sentence in (16) is ambiguous between the wide and narrow scope reading of the subject with respect to the object.

(16) Something devoured everyone.

a. There is something that devoured everyone. (something > everyone)

b. For each person, there is something that devoured him. (everyone > something)

The sentence has the single derivation as in (17).

\[
\begin{array}{c}
\text{(something,devoured,everyone):t} \\
(\emptyset,\text{devoured,everyone}):+q t,\text{something}:-q \\
(\emptyset,\text{devoured,everyone}):+k +q t,\text{something}:-k -q \\
\emptyset := \Rightarrow v +k +q t \quad \langle \text{everyone,devoured,[]} \rangle : v,\text{something}:-k -q \\
\emptyset := \Rightarrow v +k +q t \quad \langle \text{everything,devoured,[]} \rangle : v,\text{something}:-k -q \\
\emptyset := \Rightarrow v +k +q t \quad \langle \text{everyone,devoured,[]} \rangle : v,\text{something}:-k -q \\
\emptyset := \Rightarrow v +k +q t \quad \langle \text{everyone,devoured,[]} \rangle : v,\text{something}:-k -q \\
\end{array}
\]

The derivation in (17) is a underspecified semantic representation of the two possible readings of the sentence. We calculate the narrow scope reading of the subject as follows. First, we merge devour (a function from individuals to predicates) and everyone (a function from predicates to assignments) by function application and store the denotation of everyone (14e). The resultant denotation of the VP devour everyone is devour(x₀) with the function G(everyone)(λ₀)
in store. Next, we merge (by function application (14c)) the voice head with the derived expression. We then move everyone to check its -k feature without retrieving its denotation from the store (14f). In the next step, we merge someone. Now, we can retrieve the stored denotation of everyone (14g) yielding everyone(λ₀(someone(devour(x₀))))).

To calculate the wide scope reading of the subject we, first, merge devour and everyone and store the denotation of everyone. We then merge the voice head and move everyone to check its -k feature. Now, we merge something with the derived expression and store its denotation by feeding the VP another variable yielding devour(x₀)(x₁) and the functions G(something)(λ₁) and G(everyone)(λ₀) in store. In the next step, we retrieve the denotation of everyone yielding everyone(λ₀(devour(x₀)(x₁))) and G(something)(λ₁) in store. Finally, we retrieve the denotation of something and yield something(everyone (λ₀(devour(x₀)(x₁))))).

3 Scope ambiguity and coordination

It has been observed, as early as Rooth and Partee (1982), that disjunction shows properties of a scope-bearing element. In this section, we discuss an instance of scope ambiguity in coordinate structures with disjunction. We talk about different readings in gapping constructions embedded under a modal verb.

3.1 Background on gapping

Starting with Ross (1970), sentences such as (18) have been referred to as gapping. In (18), the verb ate in the second conjunct is omitted but it is interpreted as if it were there.

(18) Some ate natto and others rice.

In a gapping construction, a verb and other material can go unpronounced if their content can be recovered from the preceding conjunct. In the example (18), the verb ate of the first conjunct is the antecedent for the gap in the second conjunct. In case only a verb is gapped, the gap is called a single gap (19a). When more material is gapped, the gap is referred to as a complex gap (19b).

(19) a. Some ate natto and others rice. (single gap)
   b. Some ate the natto hungrily and others timidly. (complex gap)

Gapping can target finite verbs (20a), or finite auxiliaries or modals (20b). In the latter case, the main verb may retain in the second conjunct. In the paper, we will primarily be dealing with gapping structures such as (20b).

(20) a. Jill watched the hockey game and Jori the luge race.
   b. Jill will refereee the hockey game and Jori time the luge race.

3 These sentences are from Lin (2002) (p.10).
In the theory of gapping, there are two main questions with regard to properties of gapping constructions. First question is about the size of the conjunct containing the gap. There are two main approaches to the size question. On the one hand, it is assumed that the conjunct containing the gap is much larger than it appears on the surface and that it is of the size of the ungapped conjunct. This approach is usually referred to as the large-conjunct approach (Ross, 1967; Neijt, 1979; van Oirsouw, 1987; Wilder, 1994, 1997; Hartmann, 2000). On the other hand, it is hypothesized that the gapped conjunct is smaller than its ungapped counterpart. This approach has been called the small-conjunct approach to gapping (Johnson, 1996, 2009; Coppock, 2001; Lin, 2002). Second question asks how the gap is produced. There are three approaches to the way the gap in the second conjunct is derived. According to the first approach, the gap is the result of ellipsis (Coppock, 2001). According to the second approach, the ‘shared’ material in gapping constructions moves across-the-board (Johnson, 2009). Third approach assumes that the gap is a null pro-form (Williams, 1997). In this paper, we will be dealing with the size question.

According to the large-conjunct approach, bigger phrases, such as TPs, are coordinated (21).

\[
\begin{array}{c}
\text{TP} \\
\text{TP}_1 \quad \text{ConjP} \\
\text{Conj} \text{TP}_2
\end{array}
\]

Some kind of a (syntactic) reduction mechanism derives the gap, by which the verb and other material of the second conjunct get deleted under identity with material in the first conjunct. Correspondingly, the sentence in (22a) receives the parse as in (22b), where the strike-out represents reduced material.

(22) a. John ate natto and Bill rice.
   b. \([TP\text{John ate natto}] \text{ and } [TP\text{Bill ate rice}]\)

The large-conjunct approach predicts that no item of the first conjunct will be able to bind an element or to scope over an element of the second conjunct. However, the prediction is not born out. The following scope and binding facts pose a problem for the large-conjunct approach (Siegel, 1984, 1987; Oehrle, 1987; McCawley, 1993; Johnson, 1996; Lin, 2002). In gapping, the subject of the first conjunct binds the pronoun in the subject of the second conjunct (23).

(23) a. No woman, can join the army and her girlfriend the navy.
   b. Not every student, bought a hat, and her brother a sweatshirt.

Standard assumptions about how binding works suggest that in the sentences (23), the subject of the first conjunct c-commands the subject of the second conjunct.
junct. Notice that binding is not possible in corresponding non-gapped sentences (24).

(24) a. *[TP No woman, can join the army] and [TP her, girlfriend can join the navy.]
    b. *[TP Not every student, bought a hat] and [TP her, brother bought a sweatshirt.]

In (24), the whole sentences (TPs) are coordinated and a quantifier of the first conjunct cannot bind into the second conjunct. On the large-conjunct approach, the sentences in (23) are analyzed as conjoined TPs and are wrongly predicted to be ungrammatical.

In the gapping sentence in (25a), the negated modal takes wide scope with respect to coordination, receiving the non-distributed modal reading, paraphrased in (25b).

(25) a. Ward can’t eat caviar and Mary eat beans.
    b. It is not possible that Ward eats caviar and Mary eats beans.

The corresponding non-gapped sentence, conjoining two TPs, has the distributed modal reading.

(26) [TP Ward can’t eat caviar] and [TP Mary can’t eat beans.]

On the large-conjunct approach, (25a) is analyzed as (26), but they do not mean the same thing.

According to the small-conjunct approach, smaller phrases are conjoined and “shared” material lies outside coordination (27).

(27) TP
    vP
    vP_1 ConjP
    Conj vP_2

According to this approach, the sentence in (28a) is a vP-coordination and has a parse as in (28b).

(28) a. John can eat natto and Bill eat rice.
    b. John, can [vP eat natto] and [vP Bill eat rice]

The approach makes correct predictions about wide scope modals and cross-conjunct binding. According to this approach, finite auxiliary and modal verbs lie outside coordination, as in (28b). This allows the modals or other auxiliary operators to take scope over the coordination. The subject of the first conjunct moves out of its vP and c-commands the subject of the second conjunct correctly predicting the binding fact, as in (29).

(29) a. No woman, can join the army and her, girlfriend the navy.
b. No woman can join the army and her girlfriend the navy.

The wide scope of modals and cross-conjunct binding facts show that a small-conjunct approach should be adopted to analyze gapping constructions in English (Coppock, 2001; Lin, 2002; Johnson, 2009).

3.2 Scope of modals in gapping with disjunction

In English, gapping with disjunction when embedded under a modal verb has more than one reading (Hulsey, 2008). Consider, for instance, the gapping sentence in (30) which is ambiguous between the wide and narrow scope readings of the modal must. According to the first reading, the modal takes wide scope over the entire disjunction (30a). On this reading, the sentence has one requirement that both characters do not weigh the same, as paraphrased in (30a). According to the second reading, the modal takes narrow scope with respect to disjunction. On this reading, the modal distributes into each disjunct and the sentence denotes two requirements that The Incredible Hulk must outweigh the Thing and the Thing must outweigh the Hulk, as indicated by the continuation . . . , but I don’t remember which in (30b).

(30) The Incredible Hulk must outweigh the Thing or the Thing outweigh the Hulk.

  a. They must not weigh the same. (must > or)
  b. But I don’t remember which. (or > must)

Hulsey (2008) has observed that under the small-conjunct approach of gapping the narrow scope reading of the modal is not predicted, which is puzzling. On this approach, the gapping sentence in (30) has a parse as in (31). Here, the modal lies outside the disjunction and is predicted to always take wide scope.

(31) $[\text{TP}_P \text{The Incredible Hulk}, \text{must } [\text{v}_P [\text{v}_P \text{outweigh the Thing}] \text{ or } [\text{v}_P \text{the Thing outweigh the Hulk}]]$

Recently, an alternative semantics originally proposed for questions in English (Hamblin, 1973) has been extended to natural language quantification (Ramchand, 1997; Hagstrom, 1998) including indeterminate phrases (Kratzer and Shimoyama, 2002; Shimoyama, 2006) and disjunction (Alonso-Ovalle, 2006; Hulsey, 2008). The key idea behind the alternative semantics is that linguistic items of the different categories have “denotation-sets” rather than denotations. For example, the proper name ‘Mary’ stands not for the individual ‘Mary’ but for the set whose only member is ‘Mary’. Similarly, indeterminate phrases and disjunction denote sets whose members are Hamblin alternatives created by an indeterminate phrase and by disjuncts, respectively. For example, a disjunction phrase ‘Mary or John’ is a set where ‘Mary’ and ‘John’ are two members of the set. The new alternative-based semantics approach can account for the disjunction puzzle (Hulsey, 2008). According to the new approach, disjunction denotes a set of Hamblin alternatives rather than the logical $\lor$ operator. The alternatives introduced by or are caught by an existential closure operator $\exists$ triggered
under the scope of a modal verb. Scope of the disjunction is a point of existential closure application. According to this approach, when the existential closure applies before the modal verb, the sentence has wide scope reading. Narrow scope reading of the modal is derived by first combining the modal with the disjunction phrase using the function application rule and then closing the set with existential closure (for more details see Hulsey, 2008). The two derivations of wide and narrow scope readings are schematized in (32).

(32) \[ T P \text{The Incredible Hulk, must } [ v_P [ v_P, \text{outweigh the Thing} \text{ or } v_P \text{the Thing outweigh the Hulk.}]] \]
   a. \( \exists (\text{the IH outweigh the T} \text{ or } \text{the T outweigh the H}) \)
   b. \( \exists ((\text{must(\text{the IH outweigh the T}})) \text{ or } (\text{must(\text{the T outweigh the H}}))) \)

In the next section, we implement the solution to the puzzle using compositional semantics of Minimalist Grammars (Kobele, 2006). According to the approach, denotation of a disjunction phrase can be put on store in the process of derivation and retrieved later for interpretation. We derive narrow scope reading of the modal by storing the disjunction phrase and by allowing the modal to distribute over each disjunct (by function application). We derive wide scope reading of the modal in a regular way, without putting the disjunction phrase on store. Implemented in such a way, the approach captures a set-forming property and does not abandon the standard semantics of disjunction. The approach derives scope ambiguities in coordination and formalizes an alternative-based semantics approach to disjunction (Alonso-Ovalle, 2006; Hulsey, 2008). It sheds light on a question whether alternative semantics (Hamblin, 1973) might be formalized.

4 Deriving scope effects with Minimalist Grammars

In this section, we apply the direct compositionality and MGs approach to derive the scope of modals in gapping with disjunction. Recall, that the sentence in (33) is ambiguous between the wide and narrow scope readings of the modal with respect to disjunction (Hulsey, 2008).

(33) *For the Red Sox to make the playoffs...* (context)
    The Sox must beat the Yankees or the Angels lose to the Mariners.
    a. Either of two events is sufficient. (must > or)
    b. ... , but I don’t remember which. (or > must)

According to the first reading, the modal takes wide scope and the sentence has the denotation that *for the Red Sox to make the playoffs, either of two events is sufficient* (33a). According to the second reading, the modal takes narrow scope and the sentence can be continued with ... , but I don’t remember which (33b).

Following previous research on syntax of disjunction (Larson, 1985; Higginbotham, 1989), I assume that *either* marks the left edge of a disjunction. In the grammar (34), the idea is implemented as an existential closure phrase \( \exists P \), which hosts *either* or a phonologically null scopal element *op* in its specifier position.
sox::n yankees::n angels::n mariners::n
the::=n d the::=n d -k to::=d p beat::=d =d v lose::=p =d v must::=∃ +k +q t ε::=v =q ∃ op::q -q
or::=v conj v<<conj startCategory(t)

The MG parser and grammar in (34) output a tree structure in (35).

The tree structure (35) is an underspecified semantic representation of the two possible readings of the gapping sentence. The sentence has the single derivation, which involves the following (relevant) derivation steps.

1. merge(or::=v conj, v)
2. merge(1, v)
3. merge(ε::=v =q ∃, 2)
4. merge(3, op::q -q)
5. merge(must::=∃ +k +q t, 4)
6. move(5)
We calculate the modal wide scope reading of the sentence (33a) as follows. We assume that denotations of each disjunct are calculated in a regular way. Then, we merge disjunction with the first and the second vP disjuncts by function application mode of semantic combination (14a). At this point, we do not store the denotations of the disjuncts. By function application (14a), we merge the existential closure phrase and the modal. We move the subject to satisfy the -k feature (by Id mode of semantic combination (14f)). No store of the disjunction phrase is required to derive the wide scope reading of the modal.

We calculate the modal narrow scope reading of the sentence (33b) in the following way. We merge disjunction with the first vP disjunct, assigning the disjunction an index or(x₀) and putting the denotation of the first disjunct in store G(the angels lose to the mariners)(λ₀). We use the Store mode of semantic combination (14e). In the next step, we merge the derived expression with the second disjunct or(x₀)(x₁) and put it in store G(the sox beat the yankees)(λ₁), G(the angels lose to the mariners)(λ₀). By function application (14c), we merge the existential closure phrase and the modal must(or(x₀)(x₁)), with both disjuncts still being in store. We move the subject to satisfy the -k feature (by Id mode of semantic combination (16a)). Finally, we move the phonologically null scopal element and retrieve the disjuncts from the store (by Retrieve mode of semantic combination (16a)) must(or(x₀)(x₁))(G(the sox beat the yankees)(λ₁))(G(the angels lose to the mariners)(λ₀)), receiving a distributed modal interpretation of the sentence.

5 Summary and outlook

In this paper, we addressed scope ambiguity in coordination from a computational point of view. We adopted a Minimalist Grammars formalism (Stabler, 1997; Stabler and Keenan, 2003; Kobele, 2006), which uses underspecification in semantic representations. The approach derives scope ambiguities in coordination and formalizes an alternative-based semantics approach to disjunction (Alonso-Ovalle, 2006; Hulsey, 2008). It sheds light on a question whether alternative semantics (Hamblin, 1973) might be formalized.
Bibliography


