A Discourse Copying Algorithm for Ellipsis and Anaphora Resolution

Andrew Kehler
Harvard University
Aiken Computation Laboratory
33 Oxford Street
Cambridge, MA 02138
kehler@das.harvard.edu

Abstract

We give an analysis of ellipsis resolution in terms of a straightforward *discourse copying* algorithm that correctly predicts a wide range of phenomena. The treatment does not suffer from problems inherent in identity-of-relations analyses. Furthermore, in contrast to the approach of Dalrymple et al. [1991], the treatment directly encodes the intuitive distinction between full NPs and the referential elements that corefer with them through what we term *role linking*. The correct predictions for several problematic examples of ellipsis naturally result. Finally, the analysis extends directly to other discourse copying phenomena.

1 Introduction

A simple example of verb phrase (VP) ellipsis is given in sentence (1):

(1) John likes his mother, and Bill does too.

The stranded auxiliary in the second clause indicates the deletion of a verb phrase, the meaning of which is to be determined from another clause, in this case, the first clause. Following Dalrymple, et al. [1991], we term the clause that the VP is copied from the *source* clause, and the clause which contains the elided VP the *target* clause. As is well established in the literature, *referential elements*[1] in the source clause VP may give rise to *strict/sloppy* ambiguities in the target clause; for example in sentence (1) the target clause may receive one of two readings, namely one in which Bill likes *John's* mother (the *strict* reading), and one in which Bill likes *his own* mother (the *sloppy* reading).

In this paper we address the distribution of *strict* and *sloppy* readings in VP-ellipsis and other anaphoric phenomena. In Section 2 we discuss problems with past approaches, as well as support our decision to treat ellipsis resolution as a primarily semantic process. In Section 3 we give a representation for events that exhibits what we term *role linking* between referential elements and their referents, and describe an algorithm for ellipsis resolution that uses this representation. Section 4 demonstrates how use of this representation and algorithm results in the correct analysis of several problematic cases of ellipsis without appeal to additional constraints. In Section 5 we argue that the strict/sloppy distinction is not a property of ellipsis alone, but is instead inherent in what we term the *discourse copying* process. Several discourse copying phenomena at the event and NP level are shown to be accounted for by our algorithm. We conclude in Section 6.

2 Past Approaches

Much attention has been paid to the ellipsis problem in linguistics (e.g., [Dahl, 1972; Dahl, 1974; Fiengo and May, 1990; Gawron and Peters, 1990; Haïk, 1987; Hellan, 1988; Klein, 1987; Lappin, 1992; Sag, 1976; Williams, 1977]), as well as in natural language processing (e.g., [Dalrymple *et al.*, 1991; Hardt, 1992; Lappin and McCord, 1990; Prüst *et al.*, 1991]). We begin by briefly pointing out several problems with some of these approaches.

Syntactic accounts of ellipsis (e.g., [Fiengo and May, 1990; Haïk, 1987; Hellan, 1988; Lappin, 1992;

[1] For present purposes, we will use the term *referential elements* as synonymous with *pronouns*. In Section 5 we discuss how this class may be extended to include implicit arguments as well.

Lappin and McCord, 1990]) posit the copying of syntactic structure from the source clause representation to the target clause representation.[2] Such accounts fail to explain certain empirical facts. First, an active voice target clause can obtain its representation from a passive voice source clause (and vice versa), but in these cases there is no surface syntactic VP constituent to copy:

(2) A lot of this material can be presented in a fairly informal fashion, and often I do. (from text of [Chomsky, 1982])

(3) This problem was to have been looked into, but obviously nobody did. (Vincent Della Pietra, in conversation)

Second, an elliptical clause can obtain its referent from an event representation evoked into the discourse world by a nominalization:

(4) Meanwhile, they sense a drop in visitors to the city. Those who do, they say, are not taking cabs. (Chicago Tribune, courtesy Gregory Ward)

Third, an elliptical clause may have multiple antecedents:

(5) Mary wants to go to Spain and Fred wants to go to Peru, but because of limited resources, only one of them can. ([Webber, 1978])

Several semantic accounts of ellipsis utilizing logical forms have been proposed. Following Dalrymple et al. [1991], we distinguish between *identity-of-relations* analyses (e.g., [Gawron and Peters, 1990; Klein, 1987; Sag, 1976; Williams, 1977]) and *non-identity* analyses [Dalrymple *et al.*, 1991]. Identity-of-relations analyses treat source clauses as having ambiguous *derivations*, with target clauses receiving one such derivation. For example, the source clause in example (1) would have derivations that (at some level) lead to the following two interpretations:

John has the property P where:

(6) P = likes *John's* mother
(7) P = likes *his own* mother

These two properties lead to the same reading for the source clause. However, under an identity-of-relations analysis, if the target receives a strict interpretation, then necessarily (6) is the correct interpretation for the source, whereas if the target receives a sloppy reading, then (7) is necessarily the correct interpretation.

The following example, from Dahl [1972], is termed as a case of *cascaded ellipsis* in Dalrymple, et al. [1991], and is problematic for identity-of-relations analyses:

(8) John realizes that he is a fool, but Bill_i does not, even though his_i wife does.

An acceptable, and perhaps preferred, reading for (8) is:

(9) John realizes that *John* is a fool, but Bill does not realize that *Bill* is a fool, even though Bill's wife realizes *Bill* is a fool.

Example (8) contains two cases of ellipsis; the reading in (9) results from the second clause receiving a *sloppy* interpretation from the first and the third clause receiving a *strict* interpretation from the second. An identity-of-relations analysis, however, specifically predicts that the reading given in (9) does not exist. Because the second clause will have the sloppy derivation received from the first, the strict derivation that the third clause requires from the second will not be present.

Dalrymple, et al. [1991] (henceforth DSP) give an account of ellipsis resolution based on an equational analysis utilizing higher-order unification. Unlike identity-of-relations accounts, no unmotivated ambiguity is predicated to exist for VPs with pronouns, instead the ambiguity exists in the *resolution* process. As a result, reading (9) for sentence (8) is correctly predicted to exist.

We step through DSP's analysis of example (1) to acquaint the reader with their system. The representation of the source clause in this example is:

(10) $likes(John, mother\_of(John))$

The ellipsis is resolved by deriving a property P such that representation (10) results from applying it to $John$:

(11) $P(John) = likes(John, mother\_of(John))$

This derivation yields two solutions:[3]

(12) $P = \lambda x.likes(x, mother\_of(John))$
(13) $P = \lambda x.likes(x, mother\_of(x))$

To generate a reading for the target clause, either one of these representations may be applied to the element in the target clause that is parallel to *John*, in this case *Bill*.[4]. Applying relation (12) to Bill yields the strict reading in (14); applying relation (13) yields the sloppy reading given in (15):

(14) $P(Bill) = likes(Bill, mother\_of(John))$
(15) $P(Bill) = likes(Bill, mother\_of(Bill))$

While the DSP account is comprehensive, some flaws remain. First, they claim that their analysis covers cases of *stripping* as well as ellipsis. Consider the case of stripping in example (16):

[2] Fiengo and May [Fiengo and May, 1990] actually posit copying of LF representations, but their analysis shares the problems of the syntactic accounts.

[3] Strictly speaking, there are four solutions to equation (11). DSP employ a notion of *primary occurrence* to constrain property derivation to only these two solutions. This point is tangential to our purposes; the interested reader should consult Dalrymple et al. [1991].

[4] The DSP analysis separates the processes of parallel element determination and ellipsis resolution. Their algorithm models the latter process.

(16) John likes his mother, and Mary's too.

This sentence has only the following reading:

(17) *John likes John's mother, and John likes Mary's mother.*

The representation for the source clause in DSP's system is given by:

(18) $likes(John, mother\_of(John))$

After deriving two possible properties P, either of the two readings given in (19) and (20) may be derived:

(19) $P = \lambda x.likes(John, mother\_of(x))$

$P(Mary) = likes(John, mother\_of(Mary))$

Reading: *John likes Mary's mother*

(20) $P = \lambda x.likes(x, mother\_of(x))$

$P(Mary) = likes(Mary, mother\_of(Mary))$

Reading: *Mary likes Mary's mother*

However, only reading (19) actually exists for sentence (16), so DSP's system overgenerates in this case. We return to this example in Section 4.3.

Second, to account for *missing reading* examples and the *5-reading* sentence, DSP postulate an unspecified "suitable definition of generalized antecedent linking", and need to impose an additional constraint on property derivation requiring that when an element is abstracted over, so must its generalized antecedent. These examples are discussed in Sections 4.4 and 4.5. Third, their account does not extend cleanly to similar phenomena at the noun phrase level, a topic we return to in Section 5.

3 An Event Representation and Algorithm

We seek an analysis of ellipsis that preserves the advantages of the DSP analysis but remedies the problems we have noted. The following sections describe the event representation and the resolution algorithm that embody our analysis.

3.1 Event Representation

To highlight the general mechanism of our approach we will utilize a simple, Davidsonian-style data structure where events are reified as variables [Davidson, 1967]. For example, sentence (21) will be represented by the feature-based event structure in (23); this corresponds to the more standard logical form given in (22)[5]:

(21) John sees Mary

(22) e_1: see(e_1), agent(e_1, John), theme(e_1, Mary)

(23) e_1: [predicate: see
agent: John
theme: Mary]

An additional requirement that we impose on the representation is what we term *role linking*. In order to link referential elements to their referents, functions are used to retrieve the value of roles in event structures. For example, the function *agent(e_1)* would be used to link a referential element to *John* in the representation of e_1 in (23), likewise *theme(e_1)* would be used to link to *Mary*. Consider sentence (24) and its event structure (25):

(24) John$_i$ likes his$_i$ mother

(25) e_2: [predicate: likes
agent: John
theme: [obj: mother
poss: agent(e_2)]]

The representation of *his* is a function bearing the role of the entity it refers to in the sentence. In contrast to the DSP approach, the representation for a full NP will appear in an event structure only when it is explicitly mentioned in the discourse, e.g., *John* would have appeared twice in (25) only if sentence (24) were *John likes John's mother*.

When there are several referential elements in a clause that are coreferential, each has several potential antecedents with which role linking may be established. The following rule governs role linking in this case:

(26) A referential element is linked to the most immediate coreferential element that c-commands it in the syntax.

Because we encode the distinction between full NPs and the referential elements that refer to them, we naturally account for the *stripping* example, as discussed in Section 5.3. The fact that we link referential elements through the role that their referents have in their events, in conjunction with rule (26), allows us to account for the *missing readings* examples discussed in Section 5.4 and the *5-reading* example discussed in Section 5.5 without appeal to any additional constraints on the algorithm. We describe this algorithm in the next section.

3.2 Resolution Algorithm

The discourse copying algorithm utilized by ellipsis resolution is summarized as follows:

1. Identify the source clause representation and formulate a parallel structure with unfilled roles and modifiers for the target.
2. Fill roles with entities given in the target clause.
3. Copy remaining empty role fillers from the source:
   (a) Identify *parallel elements*, i.e., the objects in the source representation corresponding

[5] We use semantic role names like *agent* and *theme* somewhat loosely.

to the empty roles in the target.[6]

(b) All role fillers may be (i) *referred to*, where the appropriate function is used to link the role filler to the corresponding object in the source representation. In the case that the role filler is a function *with a link to the source event*, it may also be (ii) *copied*, where a new instantiation of the function is created and the source event variable is replaced with its corresponding parallel target event variable.

As with the DSP analysis, this discourse copying method does not assume any ambiguity in the source clause. Ambiguities are generated by the choice given in Step 3b; *referring* will result in a strict reading and *copying* will result in a sloppy reading.

4 Treatment of Phenomena

We show how our algorithm accounts for a variety of ellipsis phenomena. Section 4.1 illustrates that a basic case of strict and sloppy ambiguity is correctly analyzed. Section 4.2 shows that the algorithm accounts for the *cascaded ellipsis* case, thereby retaining the advantages of the DSP approach over identity-of-relations analyses. The remaining subsections show how our algorithm accounts for the cases cited as problematic for DSP in Section 2. Section 5 will then move beyond VP-ellipsis to discuss how related non-elliptical phenomena are accounted for.

4.1 Ellipsis

We consider example (1) again, renumbered as (27):

(27) John likes his mother and Bill does too.

The representation for the source clause in (27) is:

(28) e_1: [predicate: like
agent: John
theme: [obj: mother
poss: agent(e_1)]]

The parallel event for the target is constructed (Step 1), and *Bill* is added as the agent (Step 2):

(29) e_2: [predicate:
agent: Bill
theme:]

Step 3b can only *refer to* the value of the *predicate* role.[7] Since the theme of the source event contains a referential link to the source event itself, Step 3b allows the theme to either be *referred to* with a function or *copied* by creating a new instantiation of the function occupying the theme and replacing the event variable e_1 with its parallel event variable e_2. Referring to the theme role of e_1 yields the strict reading in (30):

(30) e_2: [predicate: like
agent: Bill
theme: theme(e_1)]

Reading: *Bill likes John's mother*

Conversely, copying the theme role of e_1 yields the sloppy reading in (31):

(31) e_2: [predicate: like
agent: Bill
theme: [obj: mother
poss: agent(e_2)]]

Reading: *Bill likes Bill's mother*

Note that if the *his* in sentence (27) refers to the mother of someone in external discourse, say *agent(e_0)*, then Step 3b must leave this intact since the link is not to the source event. This correctly yields the only available reading for the target clause in this case.

4.2 Cascaded Ellipsis

We show how our analysis accounts for the cascaded ellipsis case discussed in Section 2, repeated below:

(32) John realizes that he is a fool, but Bill$_i$ does not, even though his$_i$ wife does.

In particular, we work though the derivation of example (32) that leads to reading (33), the reading that is problematic for identity-of-relations analyses:

(33) John realizes that *John* is a fool, but Bill does not realize that *Bill* is a fool, even though Bill's wife realizes *Bill* is a fool.

The representation for the first clause in (32) is given in (34):

(34) e_1: [predicate: realize
agent: John
theme: e_2: [predicate: be
agent: agent(e_1)
be_pred: fool]]

The clause represented by (34) is the source for the elided second clause. To obtain the desired reading, Step 3b chooses the *copy* option to yield the sloppy reading. A new function is instantiated from the function in the agent role of e_2, and the event variable e_1 is replaced with its parallel event variable e_3:

[6] We follow the DSP analysis in distinguishing the process of determining parallel elements from the process of performing resolution. For all of the examples considered in this paper, one can assume that parallel elements share the same thematic role in their respective events. The reader is advised to consult Dalrymple et al. [1991] for arguments on why this constraint should be broadened.

[7] Step 3a of the algorithm as stated requires that a function be used when *referring*, i.e., technically we should fill the *predicate* role of e_2 with the function *predicate(e_1)*. To improve readability, when the object being referred to contains no links to any event, we will opt to reproduce the value in the role of the target representation rather than use a function.

(35) e_3: [predicate: realize
polarity: negative
agent: Bill
theme: e_4: [predicate: be
agent: agent(e_3)
be_pred: fool]]

Reading: *Bill does not realize that Bill is a fool*

The clause represented by (35) is the source for the elided third clause. In this case Step 3b chooses to *refer to* the agent of e_4 using a function, deriving the strict reading:[8]

(36) e_5: [predicate: realize
agent: [obj: wife
poss: agent(e_3)]
theme: e_6: [predicate: be
agent: agent(e_4)
be_pred: fool]]

Reading: *Bill's wife realizes that Bill is a fool*

Thus, the reading not derivable by identity-of-relations analyses is derivable by our algorithm.

4.3 Stripping

We now show how our analysis accounts for the *stripping* example discussed in Section 2, repeated below:

(37) John likes his mother, and Mary's too.

In the DSP analysis it is possible to derive the following non-existent reading:

(38) *John likes John's mother, and Mary likes Mary's mother.*

Our algorithm generates only the correct reading for sentence (37). The representation for the source clause in example (37) is:

(39) e_1: [predicate: like
agent: John
theme: [obj: mother
poss: agent(e_1)]]

To derive a representation for the target clause, we create a parallel event structure (Step 1) and fill in the parallel element representation for Mary (Step 2). The remaining empty role fillers are that for the predicate and agent roles. Since neither contains a link to the source event, Step 3b only has the option of referring to it:

(40) e_2: [predicate: like
agent: John
theme: [obj: mother
poss: Mary]]

Reading: *John likes Mary's mother*

This yields the correct interpretation for the target clause. Because *Mary* is parallel to the only element in the source clause that contains a role link in example (37), Step 3b(ii) of the algorithm is never entered. As a result, no ambiguity exists and therefore (40) is the only derivable reading.

4.4 Missing Readings

One might expect that for source VPs which contain N referring elements, 2^N readings would be possible. However, Dahl [1974] noticed that the following example has only three readings, not four:

(41) Bill believed that he loved his wife, and Harry did too.

Out of the expected readings (42a)-(42d), reading (42d) is missing:

(42a) Harry believed that *Bill* loved *Bill's* wife.
(42b) Harry believed that *Harry* loved *Harry's* wife.
(42c) Harry believed that *Harry* loved *Bill's* wife.
(42d) # Harry believed that *Bill* loved *Harry's* wife.

Because *he* c-commands *his* in sentence (41), by rule (26) the pronoun *his* is linked to the pronoun *he* instead of directly to *Bill*. The event structure for the source clause in (41) is therefore:

(43) e_1: [predicate: believe
agent: Bill
theme: e_2: [predicate: love
agent: agent(e_1)
theme: [obj: wife
poss: agent(e_2)]]]

We show that the reading in (42d) is correctly predicted not to exist by applying the algorithm to derive all possible readings for the target clause. Step 3b will have two options for each of the role links occupying the *agent* role of e_2 and the *poss* role of the *theme* role of e_2, resulting in a total of four possibilities. In the first case both objects are *referred to*, yielding the *all-strict* reading given in (42a):

(44) e_3: [predicate: believe
agent: Harry
theme: e_4: [predicate: love
agent: agent(e_2)
theme: theme(e_2)]]

Reading: *Harry believed that Bill loved Bill's wife*

In the second case both functions are *copied*, yielding the *all-sloppy* reading given in (42b):

(45) e_3: [predicate: believe
agent: Harry
theme: e_4: [predicate: love
agent: agent(e_3)
theme: [obj: wife
poss: agent(e_4)]]]

[8] We omit the feature for polarity when it is positive, as in representation (36).

Reading: *Harry believed that Harry loved Harry's wife*

In the third case the function occupying the agent role of e_2 is *copied* and the structure occupying the theme role of e_2 is *referred to*, yielding reading (42c):

(46) e_3: [predicate: believe
agent: Harry
theme: e_4: [predicate: love
agent: agent(e_3)
theme: theme(e_2)]

Reading: *Harry believed that Harry loved Bill's wife*

These three readings are the acceptable ones for the target clause in (41). The algorithm also allows for a fourth possibility in which the function occupying the agent role of e_2 is *referred to*, and the function occupying the theme role of e_2 is *copied*:

(47) e_3: [predicate: believe
agent: Harry
theme: e_4: [predicate: love
agent: agent(e_2)
theme: [obj: wife
poss: agent(e_4)]]]

Reading: *Harry believed that Bill loved Bill's wife*

In this case the function agent(e_4) returns the value of agent(e_2), namely *Bill*, which again yields the all-strict reading in (42a). Thus, the non-existent reading given in (42d) is not derivable by the algorithm.

This behavior is a result of our choosing to link referential elements to their referents via their *roles* in the event structures. During the derivation of the representation given in (47), we chose the sloppy option for the second referential element. Instead of replacing the element itself with its parallel *element* in the event structure as other accounts do, we replaced it with a link to its parallel *role* in its parallel *event*. This process was "sloppy" in that the resulting link is to the agent of e_4 instead of to the agent of e_2; however, the resulting effect is "strict" because the agent of e_4 is the same as e_2 in this case.

The use of c-command as the linking criterion explains why many speakers get all four expected readings in (48) and (49), which are otherwise very similar to (41):

(48) Bill believed that his wife loved him, and Harry did too.

(49) Bill believed that his wife loved his brother, and Harry did too.

Since neither pronoun c-commands the other in these cases, the event structure for the source clause would have both directly linked to *Bill*, and the algorithm would derive all four readings for the target clause. Sag [1976] however notes that example (50) only has three readings:

(50) Edith said that finding her husband nude had upset her, and Martha did too.

Out of the four possibilities for the target, reading (51) is missing:

(51) # Martha said that finding *Martha's* husband nude had upset *Edith*.

The algorithm derives only the three correct readings if the first instance of *her* in sentence (50) is linked to the second. Since neither pronoun c-commands the other, this violates the linking rule. However, as noted by Reinhart [1983] (pp. 179-180), "experiencing" verbs such as *upset* often pose problems for linguistic analyses utilizing c-command. What is required for the linking rule is a notion of a *reflexive context* that applies to pronouns of all cases (for which c-command is an imperfect approximation). For instance, the NP *her husband* in sentence (50) is in a reflexive context, i.e., replacing 'her husband' with an accusative pronoun referring to Edith requires the reflexive form:

(52) ... finding herself$_i$/*her$_i$ nude had upset her$_i$.

Because the *her* in *her husband* in sentence (50) is in the reflexive context of the second *her*, the corresponding role link is required in the event structure. Thus, the algorithm works correctly for example (50); the flaw arises from using c-command in the linking rule to model reflexive contexts. In the future we expect to revise the linking rule by adopting rules superseding c-command for predicting reflexivization. The question of whether our analysis can be considered purely semantic rests on the question of whether reflexivization is syntactically or semantically controlled.

4.5 5-Reading Example

DSP discuss the following example (from Gawron and Peters [1990]) as a point of departure among previous analyses:

(53) John revised his paper before the teacher did, and Bill did too.

DSP claim that this sentence has five readings; we agree. The DSP analysis as presented derives six readings. To obtain only the correct five readings, they appeal to the constraint on abstracting over generalized antecedents mentioned in Section 2; however, a precise method for linking elements to generalized antecedents is not given. Our algorithm generates only the correct five readings without appeal to any additional constraints or processes.

The notably absent reading for the third clause in this case is given in (54):

(54) Bill revised John's paper before the teacher revised Bill's paper.

We step through the derivation of some of the readings to show that no representation for reading

(54) is derived.[9] We first derive the possible readings for the first ellipsis. The representation for the source clause is:

(55) e_1: [predicate: revise
agent: John
theme: [obj: paper
poss: agent(e_1)]]

We add the temporal modifier, parallel event structure, and role fillers for the representation of the first elided clause:

(56) e_1: [predicate: revise
agent: John
theme: [obj: paper
poss: agent(e_1)]
time: [relation: before
obj: e_2: [predicate: revise
agent: teacher
theme:]]]

The filler of the theme role in the representation for the source event contains a link to that, so there are two options. First, the theme may be *referred to*, yielding the *strict* reading:

(57) e_1: [predicate: revise
agent: John
theme: [obj: paper
poss: agent(e_1)]
time: [relation: before
obj: e_2: [predicate: revise
agent: teacher
theme: theme(e_1)]]]

Reading: *The teacher revised John's paper*

Alternatively, the theme may be *copied*, yielding the *sloppy* reading:

(58) e_1: [predicate: revise
agent: John
theme: [obj: paper
poss: agent(e_1)]
time: [relation: before
obj: e_2:
[predicate: revise
agent: teacher
theme: [obj: paper
poss: agent(e_2)]]]

Reading: *The teacher revised the teacher's paper*

We now consider the readings for the second ellipsis. Unlike the case of cascaded ellipsis, in this example the second ellipsis has the entire conjoined clause as its source. We first consider the readings derived from the strict reading represented in (57). In each of the following derivations event e_3 is parallel to event e_1, and event e_4 is parallel to e_2. After the event and non-referential role information is copied, there are four options. *Referring* to both of the roles containing links to a source event results in the *all-strict* reading:

(59) e_3: [predicate: revise
agent: Bill
theme: theme(e_1)
time: [relation: before
obj: e_4: [predicate: revise
agent: teacher
theme: theme(e_2)]]]

Reading: *Bill revised John's paper before the teacher revised John's paper*

In the second possibility, both roles can be *copied*, resulting in the *all-sloppy* reading:

(60) e_3: [predicate: revise
agent: Bill
theme: [obj: paper
poss: agent(e_3)]
time: [relation: before
obj: e_4: [predicate: revise
agent: teacher
theme: theme(e_3)]]]

Reading: *Bill revised Bill's paper before the teacher revised Bill's paper*

Third, the *poss* role of the *theme* role of e_1 may be *copied* and the *theme* role of e_2 may be *referred to*:

(61) e_3: [predicate: revise
agent: Bill
theme: [obj: paper
poss: agent(e_3)]
time: [relation: before
obj: e_4: [predicate: revise
agent: teacher
theme: theme(e_2)]]]

Reading: *Bill revised Bill's paper before the teacher revised John's paper*

Finally, the *poss* role of the *theme* role of e_1 may be *referred to* and the *theme* role of e_2 may be *copied*:

(62) e_3: [predicate: revise
agent: Bill
theme: theme(e_1)
time: [relation: before
obj: e_4: [predicate: revise
agent: teacher
theme: theme(e_3)]]]

Reading: *Bill revised John's paper before the teacher revised John's paper*

Note that the reading (62) is the same as the *strict/strict* reading in (59). Thus, the algorithm so

[9] We thank an anonymous reviewer for pointing out a flaw with a slightly different manifestation of this account in predicting the correct readings for this example.

far has generated three readings and has not generated the non-existent reading (54). We leave it to the reader to determine that when using representation (58) as the source for the final ellipsis, the following two readings are generated:

(63) *Bill revised John's paper before*
the teacher revised the teacher's paper

(64) *Bill revised Bill's paper before*
the teacher revised the teacher's paper

The algorithm therefore derives all and only the correct five readings for example (53). Recall that the algorithm's ability to avoid the non-existent reading in the *missing reading* cases in Section 4.4 was due to our choosing to link referential elements to their referents through their *roles* in the event structures. In that case, the critical point that implicitly eliminated the missing reading occurred during a derivation where the sloppy option was chosen. The ability of the algorithm to avoid generating the missing reading given in sentence (54) is also due to our role linking scheme, but in this case the crucial step was in choosing the *strict* option in the derivation of the first ellipsis (which resulted in representation (57)). Because the algorithm referred to the *theme* of e_1 through its role instead of replicating it, the representation for the non-existent reading could not be derived during the resolution of the second ellipsis. In either case, accounts that rely on parallel *elements* cannot avoid these readings without appeal to additional constraints. Our analysis requires no such constraints; the correct readings naturally result from the mechanism itself.

5 Beyond Ellipsis

The hallmark of the discourse copying process is the need not only to refer to a previously mentioned entity or event, but to create a new instantiation of it. VP-ellipsis is one such process; the meaning of the source clause serves as both the referent and the object from which a new, more general instantiation is created. Here we claim that the strict/sloppy distinction is an inherent property of the discourse copying process, and therefore not of VP-ellipsis resolution alone. Our algorithm is directly applicable to a wide variety of discourse copying phenomena at both the VP and NP levels.

All of the following reference phenomena require discourse copying, and therefore exhibit strict/sloppy ambiguities:

- VP-ellipsis: *John likes his mother, and Bill does too.*
- Pronominal Event Anaphora: *John got shot by his father. That happened to Bob too.*
- Definite Event Anaphora: *John kissed his wife, and Bill followed his example.* ([Dahl, 1972])
- 'Only': *Only John loves his mother.*
- 'One' Anaphora: *Although John bought a picture of his son, Bill snapped one himself.*

Definite NPs and pronouns referring to NPs may also require discourse copying in certain restricted contexts. In these cases, strict/sloppy ambiguities are present:

- Definite NPs: *John actually remembered his wife's birthday. Most men forget this important date.*
- 'Lazy' Pronouns: *The man who gives his paycheck to his wife is wiser than the man who spends it.* ([Karttunen, 1969])

We make a distinction between the processes of determining when discourse copying applies and performing the resolution. We show how the representations of some of these examples are resolved by our algorithm.

The pronouns *it* and *that* can be used to refer to events. Usage such as that in example (65) requires *discourse copying* since a new instantiation of an existing event needs to be applied to the target representation:

(65) John got shot by his father.
That happened to Bill too.

That is, both (66) and (67) are possible readings for (65):

(66) Bill was shot by *John's* father.
(67) Bill was shot by *Bill's* father.

These readings are generated by the algorithm in analogous fashion to the ellipsis cases. The representation for the source clause is:

(68) e_1: [predicate: shot
agent: [obj: father
poss: theme(e_1)]
theme: John]

Assuming that Bill is the *theme* of the target event, *referring* to the agent role yields the representation for the strict reading in (69), and *copying* it yields the representation for the sloppy reading in (70):

(69) e_2: [predicate: shot
agent: agent(e_1)
theme: Bill]

Reading: *Bill was shot by John's father*

(70) e_2: [predicate: shot
agent: [obj: father
poss: theme(e_2)]
theme: Bill]

Reading: *Bill was shot by Bill's father*

Dahl [1972] gives similar examples involving a form of *definite event reference*:

(71) John thinks he is smart, and Bill suffers from the same delusion.

(72) John kissed his wife. Bill followed his example.

Once the VP in the targets are recognized as event referential, the algorithm readily applies to these cases (with parallel elements being *John* and *Bill*), as it does not require any degree of syntactic or logical form parallelism between the referring expression and the referent.

'One' anaphora also requires a form of discourse copying, although for discourse entities instead of events:

(73) John bought a picture of his son, while Bill snapped *one* himself.

Most speakers find sentence (73) to be ambiguous between strict and sloppy readings, i.e., *one* could refer to a picture of *John's* son or of *Bill's* son.[10] By applying our analysis to the discourse copying of objects, these readings naturally result. The source representation for sentence (73) is given in (74):

(74) e_1: [predicate: bought
agent: John
theme: [obj: picture
arg: [obj: son
poss: agent(e_1)]]]

We construct a representation for the information given in the target sentence without the 'one'-anaphoric NP:

(75) e_2: [predicate: snap
agent: Bill
theme:]

Step 3b of the algorithm operates in the same way as it does for ellipsis; the two choices yield the strict reading in (76) and the sloppy reading in (77):

(76) e_2: [predicate: snap
agent: Bill
theme: theme(e_1)]

(77) e_2: [predicate: snap
agent: Bill
theme: [obj: picture
arg: [obj: son
poss: agent(e_2)]]]

As our analysis predicts, no sloppy reading exists if *his* refers to an intersentential discourse object (e.g., Fred).

The cases involving definite NPs and lazy pronouns can be handled in an analogous fashion to 'one'-anaphora. However, whereas 'one'-anaphora by its nature involves discourse copying, the contexts in which definite NPs and pronouns can copy as well as refer are more limited. Therefore, the challenge in handling these latter cases is in determining when discourse copying is licensed.

As a final note, we point out that the class of what we have termed referential elements can be extended to include implicit arguments as well as pronouns. For instance, consider examples (78) and (79), adapted from Partee [1989]:

(78) John went to a local bar to watch the Superbowl, and Bob did too.

(79) George drove to the nearest hospital, and Fred did too.

In sentence (78), *local* has an implicit argument that is linked to *John*, likewise for *nearest* and *George* in sentence (79). In each case there are strict and sloppy readings for the target clause; e.g., the target in example (78) can mean that Bob went to the bar local to John, or a bar local to himself. As is the case with pronouns, if the implicit argument is instead linked to external discourse (e.g., "local" interpreted as being to the speaker instead of to John in sentence (78)), then there is only one reading for the target; no sloppy reading exists.

Nominalizations can also contain implicit arguments that give rise to strict and sloppy readings, as in example (80):

(80) The CS500 final exam is tomorrow.
John fears failure and his brother does too.

Assuming that John fears his own failure, then John's brother may either fear John's failure or his own failure. Our algorithm readily handles these cases with the appropriate role linking of implicit arguments.

Not included in the class of referential elements are empty pronouns within infinitival clauses; these do not give rise to strict readings. For example, in (81)

(81) John wants to leave, and Bill does too.

there is only a sloppy reading for the target; it cannot be taken to mean that Bill wants John to leave. In our algorithm, the representation for empty pronouns will always have to be *copied.*

6 Summary

We have given an analysis of ellipsis resolution in terms of a straightforward *discourse copying* algorithm that correctly predicts a wide range of phenomena. The treatment does not suffer from problems inherent in identity-of-relations analyses. Furthermore, in contrast to the approach of Dalrymple et al. [1991], the treatment directly encodes the intuitive distinction between full NPs and the referential elements that corefer with them, thus allowing the correct analysis of the *stripping* example. Furthermore, we establish this distinction via the *role linking* scheme we have defined, which also allows

[10] Obtaining the sloppy reading may be aided by considering a relevant context, such as one where John's and Bill's families are on vacation together.

the correct predictions in the missing readings and 5-readings cases to come as a natural result. Finally, the analysis extends directly to other discourse copying phenomena. Future work includes extending the analysis to interact with quantificational phenomena.

Acknowledgments

This work was supported in part by National Science Foundation Grant IRI-9009018, National Science Foundation Grant IRI-9157996, and a matching grant for the latter from the Xerox Corporation. I would like to thank Mary Dalrymple, Barbara Grosz, Gal Kaplan, Shalom Lappin, Karen Lochbaum, Christine Nakatani, Stuart Shieber, and two anonymous reviewers for valuable comments on earlier drafts.

References

[Chomsky, 1982] Noam Chomsky. *Noam Chomsky on the Generative Enterprise.* Foris, Dordrecht, 1982.

[Dahl, 1972] Östen Dahl. On so-called "sloppy" identity. *Gothenburg Papers in Theoretical Linguistics*, 11, 1972. University of Göteborg.

[Dahl, 1974] Östen Dahl. How to open a sentence: Abstraction in natural language. *In Logical Grammar Reports, No. 12*, 1974. University of Göteborg.

[Dalrymple *et al.*, 1991] Mary Dalrymple, Stuart M. Shieber, and Fernando Pereira. Ellipsis and higher-order unification. *Linguistics and Philosophy*, 14:399–452, 1991.

[Davidson, 1967] Donald Davidson. The logical form of action sentences. In N. Rescher, editor, *The Logic of Decision and Action.* University of Pittsburgh Press, Pittsburgh, 1967.

[Fiengo and May, 1990] Robert Fiengo and Robert May. Anaphora and ellipsis. ms., City University of New York and University of California at Irvine, 1990.

[Gawron and Peters, 1990] Mark Gawron and Stanley Peters. *Anaphora and Quantification in Situation Semantics.* CSLI/University of Chicago Press, Stanford University, 1990. CSLI Lecture Notes, Number 19.

[Haïk, 1987] Isabelle Haïk. Bound variables that need to be. *Linguistics and Philosophy*, 11:503–530, 1987.

[Hardt, 1992] Daniel Hardt. VP ellipsis and contextual interpretation. In *COLING 92*, 1992.

[Hellan, 1988] Lars Hellan. *Anaphora in Norwegian and the Theory of Grammar.* Studies in Generative Grammar 32. Foris, Dordrecht, 1988.

[Karttunen, 1969] Lauri Karttunen. Pronouns and variables. In *Papers from the Fifth Regional Meeting of the Chicago Linguistics Society*, 1969.

[Klein, 1987] Ewan Klein. VP-Ellipsis in DR theory. In Groenindijk and Stokhof, editors, *Studies in Discourse Representation Theory and the Theory of Generalized Quantifiers.* Foris, 1987.

[Lappin and McCord, 1990] Shalom Lappin and Michael McCord. Anaphora resolution in slot grammar. *Computational Linguistics*, 16:197–212, 1990.

[Lappin, 1992] Shalom Lappin. The syntactic basis of ellipsis resolution. ms., IBM T.J. Watson Research Center, 1992.

[Partee, 1989] Barbara Partee. Deictic and anaphoric pieces of meaning. LSA Conference on Logic and Linguistics, Santa Cruz, 1989. July 19-21.

[Prüst *et al.*, 1991] Hub Prüst, Remko Scha, and Martin van den Berg. A formal discourse grammar tackling verb phrase anaphora. Ms., 1991.

[Reinhart, 1983] Tanya Reinhart. *Anaphora and Semantic Interpretation.* University of Chicago Press, Chicago, 1983.

[Sag, 1976] Ivan Sag. *Deletion and Logical Form.* PhD thesis, MIT, 1976.

[Webber, 1978] Bonnie Lynn Webber. *A Formal Approach to Discourse Anaphora.* PhD thesis, Harvard University, 1978.

[Williams, 1977] Edwin Williams. Discourse and logical form. *Linguistic Inquiry*, 8(1), 1977.