Semantic Interpretation of Unrealized Syntactic Material in LTAG

Olga Babko-Malaya
University of Pennsylvania
malayao@ldc.upenn.edu

Abstract

This paper presents a LTAG-based analysis of gapping and VP ellipsis, which proposes that resolution of the elided material is part of a general disambiguation procedure, which is also responsible for resolution of underspecified representations of scope.

1 Introduction

The problem of ellipsis resolution is to recover the interpretation of the elided material. For example, in (1), the elided VP is interpreted as being identical to the verb in the preceding sentence. Likewise, in the gapping structures, as shown in (2), the interpretation of a gap is being identified with the interpretation of the preceding verb.

(1) Mary likes Bill. Jane does too.
(2) Mary ate beans and others -- rice.

Whereas some approaches assume syntactic identity between the antecedent and the elided material (e.g. Fiengo and May 1994), others suggest that VP ellipsises are proforms, semantically identified with their antecedents (see Dalrymple et al 1991, Shieber et al 1996, Hardt 1993, 1999).

This paper follows semantic approaches to ellipsis resolution. It adopts the LTAG semantics of Kallmeyer and Romero 2004 and proposes that resolution of ellipsises and gaps is part of a general disambiguation procedure, which is also responsible for resolution of underspecified representations of scope.

2 LTAG Semantics with Semantic Unification

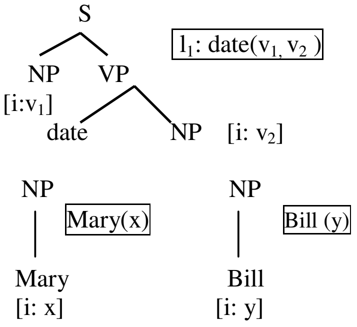
In LTAG framework (Joshi and Schabes 1997), the basic units are (elementary) trees, which can be combined into bigger trees by substitution or adjunction. LTAG derivations are represented by derivation trees that record the history of how the elementary trees are put together. Given that derivation steps in LTAG correspond to predicate-argument applications, it is usually assumed that LTAG semantics is based on the derivation tree, rather than the derived tree (Kallmeyer and Joshi 2003).

Semantic composition which we adopt is based on LTAG semantics with semantic unification (Kallmeyer and Romero 2004). In the derivation tree, elementary trees are replaced by their semantic representations and corresponding feature structures. Semantic representations are as defined in Kallmeyer and Joshi 2003, except that they do not have argument variables. These representations consist of a set of formulas (typed λ-expressions with labels) and a set of scope constraints.

Each semantic representation is linked to a feature structure. Feature structures, as illustrated by different examples below, include a feature i whose values are individual variables and features p and MaxS, whose values are propositional labels. Semantic composition consists of feature unification. After having performed all unifications, the union of all semantic representations is built.

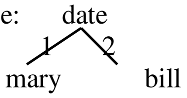
Consider, for example, the semantic representations and feature structures associated with the elementary trees of the sentence shown in (3).

(3) *Mary dates Bill*



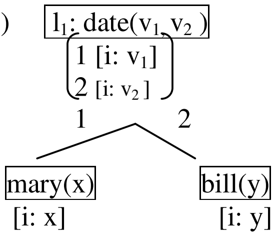
Proceedings of the 8th International Workshop on Tree Adjoining Grammar and Related Formalisms, pages 91–96,
Sydney, July 2006. ©2006 Association for Computational Linguistics

Derivation tree:



Semantic composition proceeds on the derivation tree and consists of feature unification:

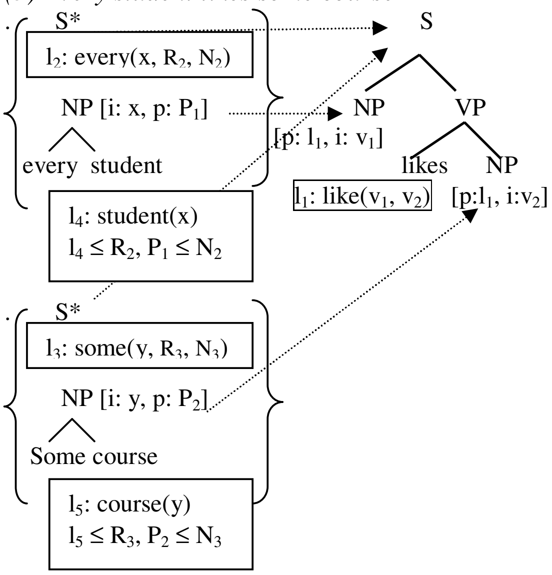
(4)



Performing two unifications, v_1=x, v_2=y, we arrive at the final interpretation of this sentence: $\boxed{l_1\text{: date(x, y), bill(y), mary(x)}}$. This representation is interpreted conjunctively, with free variables being existentially bound.

Quantificational NPs are analyzed as multicomponent TAGs, where the scope part of the quantifier introduces the proposition containing the quantifier, and the predicate-argument part introduces the restrictive clause (see Kallmeyer and Joshi 2003).

(5) Every student likes some course



Final representation

l_2: every(x, R_2, N_2)
l_4: student(x) $l_4 \leq R_2$
l_3: some(y, R_3, N_3)
l_5: course(y) $l_5 \leq R_3$
l_1: like(x, y) $l_1 \leq N_2$ $l_1 \leq N_3$

The final representation of this sentence is underspecified for scope, given that there are no constraints which restrict the relative scope of every and some. In order to obtain one of the readings, a disambiguation mapping is needed:

Disambiguations:

1. R_2 ->l_4, R_3 ->l_5, N_2 ->l_1, N_3 -> l_2:
some(y,course(y), every(x,student(x), like(x, y)))
2. R_2->l_4, R_3->l_5, N_3->l_1, N_2->l_3:
every(x, student(x), some(y, course(y), like(x, y))

Disambiguations are functions from propositional variables to propositional labels that respect the scope constraints, such that after having applied this mapping, the transitive closure of the resulting scope is a partial order.

3 The Problem of Ellipsis Resolution in LTAG semantics

Given LTAG semantics, there are two possible approaches to resolution of the elided material: reconstruction can be done as part of the unification process or as part of the disambiguation procedure. If reconstruction was done as unification, the semantic representation of the elided material would be disambiguated in the final representation. On the other hand, it is well known that resolution of ellipsises and gaps can be ambiguous. For example, the sentence in (6), discussed in Siegel 1987 and Johnson 2003 among others, has 2 interpretations:[1]

(6) *Ward can't eat caviar and his guests -- dried beans*
Can't (eat (ward, caviar)) & **eat** (his guests, dried beans))
Can't (eat(ward, caviar)) & **can't (eat**(his guests, dried beans))

As this example shows, the gap in (6) can be reconstructed by selecting either the verb or the negated modal as its antecedent. The two interpretations represent different scope readings between the conjunction and negation, which should be analyzed as underspecified in LTAG semantics. Resolution of gaps, therefore, cannot be done as part of unification, since it depends on the disambiguated interpretation. The question is whether it is possible to define an underspecified representation of these two readings, and what kind of resolution mechanism can be used to disambiguate these interpretations?

[1] Other cases of ambiguous interpretations of the elided material are discussed in section 7.

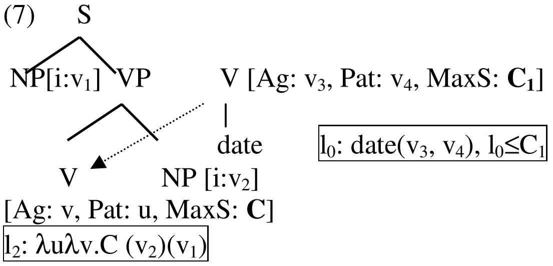
4 LTAG Semantics of Gapping

In LTAG semantics, semantic representations are introduced by lexicalized trees. In order to account for the analysis of gapping and VP ellipsis, this paper proposes that semantics should be defined on both lexicalized and non-lexicalized trees. Specifically, we propose that

Interpretation of a gap (or elided VP) is the semantic interpretation of a non-lexicalized S tree.

The semantic representations of lexicalized S trees under this new approach are derived compositionally, given the meaning of a nonlexicalized S tree and the meaning of a verb.

(7)


S
NP[i:v_1]
VP
V [Ag: v_3, Pat: v_4, MaxS: $\mathbf{C_1}$]
date
l_0: date(v_3, v_4), $l_0 \leq C_1$
V
NP [i:v_2]
[Ag: v, Pat: u, MaxS: **C**]
l_2: $\lambda u \lambda v.C\ (v_2)(v_1)$

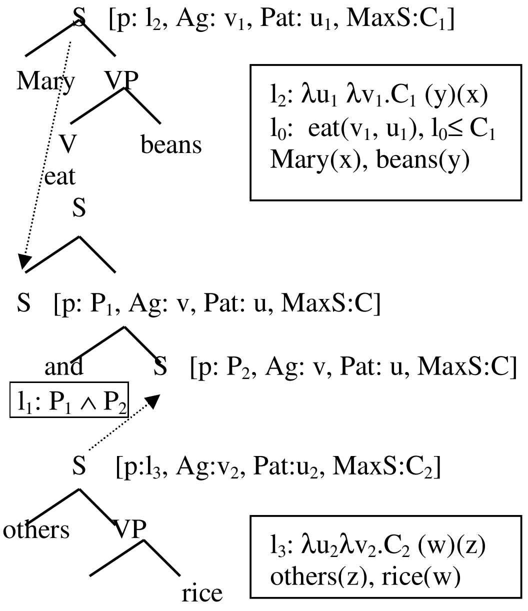
Non-lexicalized trees introduce a propositional label and a propositional variable, illustrated by l_2 and C above. If a tree is a transitive S-tree, there are two lambda bound variables, which correspond to the Agent and Patient features of the verb. Performing feature unifications (v_3=v, v_4=u,C_1=C) and scope constraint disambiguations (C->l_0), the proposition l_2 will be reduced to: $\lambda u.\lambda v.date(v, u)(v_2)(v_1)= date(v_1, v_2)$.

Given this proposal, we suggest that the semantics of gaps, VPE and other types of elided material is introduced by non-lexicalized trees. For example, the analysis of the sentence in (2) is shown in (7). Performing feature unifications (l_2=P_1, l_3=P_2, v=v_1=v_2, u=u_1=u_2, C=C_1=C_2) yields the final representation, where l_2 and l_3 are underspecified. There is only one disambiguation of the variable C in this sentence: C -> l_0, which gives us the desired interpretation of the sentence:

l_2: $\lambda u \lambda v.eat(v, u)\ (y)(x) = eat(x, y)$
l_3: $\lambda u \lambda v.eat(v, u)\ (w)(z) = eat(z, w)$

Resolution of the gap in this sentence is enforced by the feature structure of 'and', which unifies MaxS as well as Agent and Patient features. This analysis therefore accounts for the fact that gapping "is intimately entangled with the syntax of coordination (as opposed to VP ellipsis)" (Johnson 2003). On the other hand, as the next example illustrates, it is crucial that propositional variables introduced by non-lexicalized trees are not unified during semantic composition, but rather are identified with their antecedents as part of the disambiguation procedure.

(7) *Mary ate beans and others -- rice.*

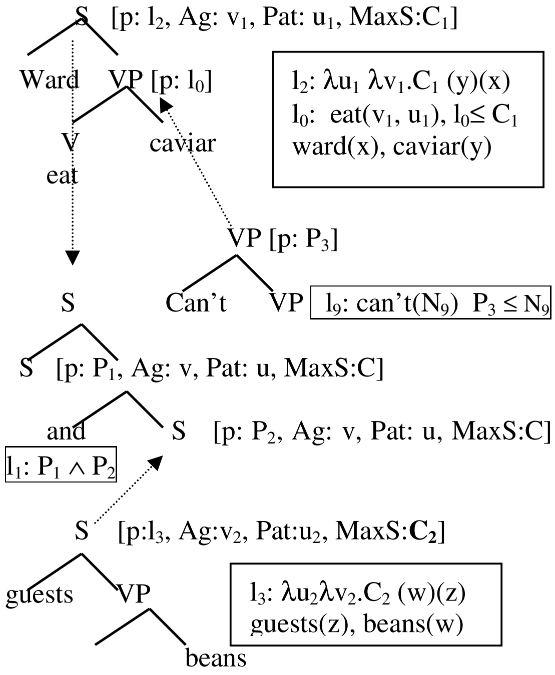

S [p: l_2, Ag: v_1, Pat: u_1, MaxS:C_1]
Mary
VP
V
beans
eat
l_2: $\lambda u_1\ \lambda v_1.C_1\ (y)(x)$
l_0: eat(v_1, u_1), $l_0 \leq C_1$
Mary(x), beans(y)
S
S [p: P_1, Ag: v, Pat: u, MaxS:C]
and
S [p: P_2, Ag: v, Pat: u, MaxS:C]
l_1: $P_1 \wedge P_2$
S [p:l_3, Ag:v_2, Pat:u_2, MaxS:C_2]
others
VP
rice
l_3: $\lambda u_2 \lambda v_2.C_2\ (w)(z)$
others(z), rice(w)

Final Representation:

l_1: $l_2 \wedge l_3$
l_2: $\lambda u \lambda v.C\ (y)(x)$ l_3: $\lambda u \lambda v.C\ (w)(z)$
l_0: eat(v, u) $l_0 \leq C$
mary(x), beans(y), others(z), rice(w)

The sentence in (8), shown below, differs from the previous one in the presence of a negated modal. The interpretation of this modal introduces a proposition l_9: can't(N_9) and a constraint $P_3 \leq N_9$. After P_3 is unified with the proposition l_0, the final representation has two constraints on the variable l_0: $l_0 \leq$ C and $l_0 \leq N_9$, and therefore two possible disambiguations. In the disambiguation 1, C is mapped to l_0, introduced by the verb 'eat', and propositions l_2 and l_3 are reduced to eat(x, y) and eat(z, w). In the disambiguation 2, the variable C is mapped to l_9, introduced by the modal, and l_2 and l_3 are reduced to can't(eat(x, y)) and can't(eat(z, w)). These disambiguations yield the desired interpretations of this sentence.

(8) *Ward can't eat caviar and his guests -- dried beans*


S [p: l_2, Ag: v_1, Pat: u_1, MaxS:C_1]
Ward VP [p: l_0]
V caviar
eat
l_2: $\lambda u_1\ \lambda v_1.C_1$ (y)(x)
l_0: eat(v_1, u_1), $l_0 \leq C_1$
ward(x), caviar(y)
VP [p: P_3]
S Can't VP
l_9: can't(N_9) $P_3 \leq N_9$
S [p: P_1, Ag: v, Pat: u, MaxS:C]
and S [p: P_2, Ag: v, Pat: u, MaxS:C]
l_1: $P_1 \wedge P_2$
S [p:l_3, Ag:v_2, Pat:u_2, MaxS:$\mathbf{C_2}$]
guests VP
l_3: $\lambda u_2 \lambda v_2.C_2$ (w)(z)
guests(z), beans(w)
beans

Final Representation

l_1: $l_2 \wedge l_3$ l_0: eat(v, u)
l_9: can't(N_9) $l_0 \leq N_9$ $l_0 \leq C$
l_2: $\lambda u \lambda v$ C (y)(x) l_3: $\lambda u \lambda v$ C (w)(z)
guests(z), beans(w), caviar(y), ward(x)

Disambiguation 1: C->l_0, N_9 ->l_1:
can't ($\underbrace{\text{eat(x, y)}}_{l_2} \wedge \underbrace{\text{eat(z, w))}}_{l_3}$

Disambiguation 2: C->l_9, N_9->l_0:
$\underbrace{\text{can't(eat(x, y))}}_{l_2} \wedge \underbrace{\text{can't (eat(z, w))}}_{l_3}$

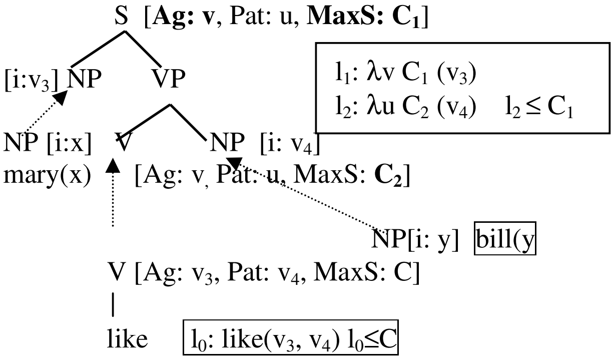
Resolution of gaps under this analysis is done as part of the scope resolution procedure on underspecified representations. A crucial feature of this analysis is that the propositions l_2 and l_3 are 'underspecified' in the final representation and the variable C is computed during the disambiguation, i.e. when all scope ambiguities are being resolved. In this respect this analysis differs from previous approaches, where the final representation did not include any variables, except for the arguments of quantifiers or other scopal elements.[2]

5 LTAG Analysis of VP Ellipsis

The analysis of gapping presented above can be easily extended to the analysis of VP ellipsis. VPE differs from gapping in that it is not restricted to coordinated structures. Whereas in the examples above resolution of gaps was enforced by the feature structure of 'and', in the case of VPE, a similar unification, forced by pragmatic constraints, results in recovering the elided material.

As the example in (9) illustrates, our analysis of VPE assumes the following modification of the semantics of non-lexicalized trees: propositions introduced by non-lexicalized trees have one lambda-bound variable, so that each argument is introduced by a separate proposition. For example, the interpretation of a transitive tree below has two propositions l_1 and l_2, and two propositional variables C_1 and C_2. The proposition l_2 corresponds to the meaning of a VP, which is missing in the standard TAG-based analyses. This decomposition of the meaning of a nonlexicalized tree, therefore, can be independently motivated by the existence of modifiers which predicate of VPs. We further assume that the MaxS feature of the S tree corresponds to the variable introduced by the agent (or the highest-ranked argument).

(9) *Mary likes Bill. Jane does too.*


S [**Ag: v**, Pat: u, **MaxS: $\mathbf{C_1}$**]
[i:v_3] NP VP
l_1: $\lambda v\ C_1$ (v_3)
l_2: $\lambda u\ C_2$ (v_4) $l_2 \leq C_1$
NP [i:x] V NP [i: v_4]
mary(x)
[Ag: v, Pat: u, MaxS: $\mathbf{C_2}$]
NP[i: y] bill(y
V [Ag: v_3, Pat: v_4, MaxS: C]
like
l_0: like(v_3, v_4) $l_0 \leq C$

Final Representation:

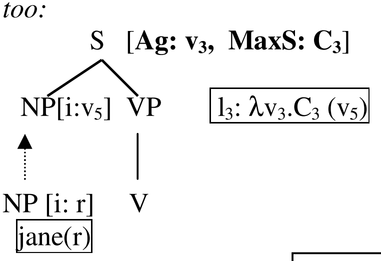
l_1: $\lambda v\ C_1$ (x)
l_2: $\lambda u\ C_2$ (y) $l_2 \leq C_1$
l_0: like(v, u), $l_0 \leq C_2$
Mary(x), Bill(y)

Applying disambiguations C_2 -> l_0, C_1 -> l_2 , we derive the following propositions:

l_2: λu.like(v, u) (y)=like(v, y)
l_1: λv.like(v, y) (x)=like(x, y)

[2] However, see Babko-Malaya 2004, where a similar analysis is proposed to account for the semantics of coordinated structures with quantified NPs.

Now consider the second sentence: *Jane does too:*



Final Representation:

$l_3: \lambda v_3\ C_3\ (r)$
Jane(r)

This sentence introduces an intransitive tree and one propositional variable C_3. This variable is not constrained within the sentence, and parallel to other pro-forms, it gets its interpretation from the previous discourse. Specifically, the interpretation of the second sentence is derived by unification of the S features of the second and the first S-trees in (9): $C_3=C_1$, $v_3=v$. Given that C_1 is mapped to l_2 above, it corresponds to the proposition being reconstructed: $C_3(=C_1)$ -> l_2

l_3: λv.like(v, u) (r) = like(r, u)

6 Scope Parallelism

Many previous approaches impose parallelism constraints on the interpretation of the elided material (e.g. Fox 2000, Asher et al 2001 among others). Under the present analysis, scope parallelism comes for free. Consider, for example, the following sentence discussed in Dalrymple et al 1991, among others, where ambiguity is resolved in the same way in both the antecedent and at the ellipsis site: *John gave every student a test, and Bill did too.* The final interpretation of the first sentence is given in (10) and has 2 possible disambiguations.

(10) John gave every student a test.

l_0: give(v, u, w)
$l_1: \lambda v.C_1\ (x)$ $l_2: \lambda u.C_2\ (y)$ $l_2 \leq C_1$
$l_3: \lambda w.C_3\ (z)$ $l_3 \leq C_2$
l_7: every(y, R_7, N_7) l_5: some(z, R_5, N_5)
l_8: student(y) l_9: test(z) john(x)
$l_0 \leq C_3$ $l_0 \leq N_5$ $l_0 \leq N_7$ $l_8 \leq R_7$ $l_9 \leq R_5$

The surface reading (every >> some) is derived by the following mapping: C_3->l_0, C_2->l_3, R_7->l_8, N_5 -> l_2, **C_1-> l_7**, R_5->l_9, N_7 -> l_5
l_2: give(v, y, z)
l_5: some(x, test(x), give(v, y, z))
l_7:every(y,student(y),some(x,test(x),give(v,y, z)))
l_1:every(y,student(y),some(x,test(x),give(x, y, z)))

The interpretation of the second sentence is derived by unifying the S-features of the S-trees (as shown in the previous section). As the result, the variables C_3 and v_3 are unified with the variables C_1 and v. Given that C_1 is being mapped to the proposition l_7 above, C_3 is being reconstructed as the proposition every(y, student(y), some(x, test(x), give(v, y, z)) and l_3 corresponds to the desired reading of this sentence:

(11) Bill did too.

l_4: Bill(r)
$l_3: \lambda v_3.C_3\ (r)$

C_3 (=C_1) -> l_7
v_3=v

l_3: λv. every(y, student(y),some(x, test(x), give(v, y, z))) (r) = every(y, student(y), some(x, test(x), give(r, y, z)))

The inverse reading (where *some>>every)* can be obtained by the following mapping C_3->l_0, C_2->l_3, R_7->l_8, N_7 -> l_2, **C_1-> l_5**, R_5->l_9, N_5 -> l_7
l_2: give(v, y, z)
l_7: every(y, student(y), give(v, y, z))
l_5:some(x,test(x),every(y,student(y),give(v,y, z)))
l_1: some(x,test(x),every(y,student(y),give(x, y, z)))

Now, when the second sentence is interpreted, C_3 is unified with C_1, which is being mapped to l_5: $C_3(=C_1)$ -> l_5. The proposition l_3, then, is reduced to: λv.some(x, test(x), every(y, student(y), give(v, y, z))) (r) = some(x, test(x), every(y, student(y), give(r, y, z)))

As this example illustrates, scope parallelism follows from the present analysis, given that C_3 is unified with a disambiguated interpretation of a VP. It can also be shown that the wide scope puzzle (Sag 1980), shown in (12) is not unexpected under this approach, however, the analysis of this phenomenon is beyond the scope of this paper.[3]

(12) A nurse saw every patient. Dr.Smith did too.
some(x, nurse(x), every(y, patient(y), see(x, y)))
*every(y, patient(y), some(x, nurse(x), see(x, y)))

[3] As Hirschbuhler 1982, Fox 2000 among others noted, there are constructions where subjects of VPE can have narrow scope relative to nonsubjects. For example, the sentence *A Canadian flag was hanging in front of every building. An American flag was too* has a reading in which each building has both an American and a Canadian flag standing in front of it. The existence of such readings does not present a problem for the present analysis, if we adopt an analysis of quantificational NPs proposed in Babko-Malaya 2004.

7 Antecedent Contained Deletion(ACD)

Further evidence for the proposed analysis comes from sentences with ACD, discussed in Sag 1980, Egg and Erk 2001, Asher et al 2001, Jacobson (to appear), and illustrated in (13):

(13) *John wants Mary to read every book Bill does.*

The elided material in this sentence is understood as either "Bill reads" or "Bill wants Mary to read". Given that 'want' and 'every' can take different scope, four possible readings are expected. However, puzzling in this case is the unavailability of one of these readings: *John wants that for every book that Bill wants Mary to read, she reads it. Let us consider the final interpretation of this sentence:

$$
\boxed{\begin{array}{l}
l_4\text{: want}(v_0, N_4) \quad l_0\text{: } \lambda v_0.C_{2\,(\text{want})}\,(r) \\
l_1\text{: read}(v, u) \quad l_2\text{: } \lambda v.C_{1\,(\text{read})}\,(x) \\
l_6\text{: } \lambda u.C_{4\,(\text{read})}\,(y) \quad l_6 \leq C_1 \quad l_8\text{: book}(y) \wedge l_3 \\
l_5\text{: every}(y, R_5, N_5) \quad l_3\text{: } \lambda v_3.C\,(z) \\
\text{mary}(x),\ \text{john}(r),\ \text{bill}(z) \\
l_1 \leq C_1,\ l_4 \leq C_2,\ l_1 \leq N_5\mathbf{,}\ l_8 \leq R_5\mathbf{,}\ l_1 \leq N_4
\end{array}}
$$

The non-lexicalized S tree introduces a proposition l_3 and variables C and v_3. These variables can be unified with either S features of the 'read'-tree (i.e. C_1 and v), or S features of the 'want'-tree (i.e. C_2 and v_0). In the first case, the small ellipsis interpretation is derived, and both scope readings are available: C = C_1, v_3 = v **C/C_1 -> l_6, C_4-> l_1**

l_2: read(x, y), l_3: read(z, y)

every >> want:

N_5 -> l_o, C_2 -> l_4, N_4 -> l_2, R_5 ->l_8

$$l_5\text{:every}(y,\ \text{book}(y)\&\underbrace{\text{read}(\mathbf{z},\ y),}_{l_3}\text{want}(r,\ \underbrace{\text{read}(\mathbf{x},\ y)}_{l_2}))$$

want >> every:

C_2 -> l_4, N_4 -> l_5, N_5 -> l_2, R_5 ->l_8

$$l_0\text{:want}(r,\text{every}(y,\text{book}(y)\&\underbrace{\text{read}(\mathbf{z},y),}_{l_3}\ \underbrace{\text{read}(\mathbf{x},\ y)}_{l_2})))$$

If C and v_3 are unified with S features of the 'want'-tree, then the large ellipsis interpretation is derived: C = C_2, v_3 = v_0, **C/C_2 -> l_4**, N_4 -> l_2, C_4 -> l_1, C_1 -> l_6

l_0: want(r, read(x, y)), l_3: want(z, read(x, y))

The reading where *every* >> *want* is derived by the following constraints: N_5-> l_0, C_1-> l_1, R_5 ->l_8

$$l_5\text{: every}(y,\ \text{book}(y)\ \&\ \underbrace{\text{want}(\mathbf{z},\ \text{read}(x,y))}_{l_3}),\ \underbrace{\text{want}(\mathbf{r},\ \text{read}(x,\ y))}_{l_0}$$

The fourth possible reading, where *want* >> *every*, however, is predicted to be unavailable under the present assumptions. This reading, *want(r, every(y, book(y) & want(z, read(x, y)), read(x, y)),* cannot be derived, since it requires the proposition l_3 to be 'inserted' within the proposition l_0.

References

Asher N., D. Hardt and J.Busquets. 2001. Discourse Parallelism, Scope, and Ellipsis", Journal of Semantics, 18(1), pp. 1-25.

Babko-Malaya O. 2004 "LTAG Semantics of NP-Coordination" in Proc. of TAG+7, Vancouver

Dalrymple M., S.Shieber and F.Pereira 1991. Ellipsis and Higher-Order Unification. In Linguistics and Philosophy 14.

Egg, M. and K.Erk 2001. A compositional approach to VP ellipsis 8th International Conference on HPSG,Trondheim, Norway

Fiengo R. and R. May 1994. Indices and Identity. MIT Press, Cambridge

Fox, D. 2000 Economy and Semantic Intepretation, MIT Press

Hardt, D. 1993. VP Ellipsis: Form, Meaning and Processing. PhD Diss. Univ. of PA

Hardt, D. 1999. Dynamic Interpretation of VP Ellipsis. Linguistics and Philosophy. 22.2. Heim, I. 2001

Jacobson, P. (to appear) Direct Compositionality and Variable-Free Semantics: The Case of Antecedent Contained Deletion", in K. Johnson (ed.), Topics in Ellipsis, Oxford University Press.

Johnson, K. 2003. In search of the Middle Field. Ms. Univ. of Mass.

Joshi A. and Y. Schabes 1997. Tree-Adjoining Grammars, in G. Rozenberg and A. Salomaa (eds.) Handbook of Formal Languages, Springer, Berlin

Kallmeyer, L. and A.K Joshi 2003. Factoring Predicate Argument and Scope Semantics: Underspecified Semantics with LTAG. Research on Language and Computation 1(1-2), 358.

Kallmeyer, L. and M. Romero. 2004. LTAG Semantics with Semantic Unification. In Proceedings of TAG+7, Vancouver, Canada

Sag, Ivan 1980 Deletion and Logical Form. Garland Press: New York

Shieber, S., F.Pereira, and M.Dalrymple 1996. Interactions of Scope and Ellipsis. In *Linguistics and Philosophy 19,* pp.527-552.