PART II

Issues in the semantics of resumptive pronouns and epithets

On the syntax and semantics of resumptive pronouns*

Edit Doron
The Hebrew University of Jerusalem

The paper presents an analysis of the contribution of resumptive pronouns to the structure and interpretation of relative clauses in Hebrew. Traditionally, resumptive pronouns have been treated as overt phonological realizations of gaps, since it was believed that the interpretation of relative clauses with optional resumptive pronouns is unaffected by whether or not they include the pronoun. Yet the paper shows that the presence of a resumptive pronoun modifies the interpretation of a relative clause. The antecedent of the resumptive pronoun must be interpreted as having wider scope than any other noun phrase within the relative clause. Thus a resumptive pronoun, but not a gap, blocks the raising of quantifiers from within the relative clause, and the de-re interpretation of noun phrases in the clause. This is accounted for by treating resumptive pronouns as pronouns, and interpreting them as resumptive only in clauses which do not contain gaps.

1. Introduction

Relative clauses in many languages have resumptive pronouns where English would have a gap. Hebrew is one such language. A conceivable way of approaching resumptive pronouns is to say that they are syntactically of the same category as gaps, and that they get the same semantic translation. The only difference would be that certain gaps get "spelled out" as pronouns. Approaches along these lines can be found in Borer (1979), Engdahl (1979) and Maling and Zaenen (1982). The same is also suggested in Gazdar (1982) and Peters (1980).

* I am grateful to Charles Kirkpatrick for many stimulating conversations on topics related to this paper, for carefully reading a previous version, and for presenteing me with detailed comments and valuable suggestions. I am grateful to Lauri Karttunen for discussing with me the data and preliminary ideas and for many helpful suggestions. I also wish to thank Akira Kurahone and Stanley Peters for related discussions.

According to the analysis I will propose here, resumptive pronouns are syntactically and semantically pronouns, and they differ in both these respects from gaps. One very simple piece of evidence in favor of my approach is that languages that make use of resumprtive pronouns use the same inventory available to them for other pronouns. Another simple fact is the following sentence (from Maling & Zaenen 1982: Footnote 20):

(1) This is the woman that John said that she and Bill are having an affair.

Acording to Maling and Zaenen, the corresponding sentences in Scandinavian languages and in Irish are good sentences. The corresponding sentence in Hebrew is also perfectly good.

The pronoun *she* in (1) is a resumptive pronoun. If it were syntactically a gap, it would be, in Gazdar's (1981) notation, of category NP/NP and therefore not conjoinable to the NP *Bill*. Under my approach, we simply have here a conjunction of two NPs.

In Section 2, I will present a fragment of Hebrew with relative clauses. In this fragment, gaps are phonological realizations of "links", whereas resumptive pronouns are undistinguishable syntactically and phonologically from other pronouns. Semantically, what gets used in the translation of a sentence in place of a gap is a variable p_i, and the meaning of the gap's antecedent is kept in a store together with the index i of the variable. Resumptive pronouns on the other hand get the same translation as other pronouns (i.e. PP{x_i}), but for the fact that the index i is also kept in a store. The rules of storage retrieval will be different for gaps and for resumptive pronouns.

In Section 3, I will show how the fragment handles syntactic and semantic differences between sentences with gaps and sentences with resumptive pronouns. Approaches that conflate gaps and resumptive pronouns would need ad hoc machinery to account for such differences.

In Section 4, I will show how the system developed in this paper accounts for the distribution of resumptive pronouns observed by Maling and Zaenen (1982) and by Engdahl (1979, 1980) in the Scandinavian languages. I will also show why my system is to be preferred to the ones proposed by these authors.

2. The fragment

The rules for a fragment of Hebrew with relative clauses are given in Appendix A. The syntactic categories used are S′ (S bar), S, VP, NP, PP, etc … I also use syntactic features such as [± tense], [± present] to account for the fact that VP complements are infinitival (cf. S3 b) and that there is a "rule of pro drop" when the VP is not in the present tense (cf. S1 b).

The grammar in Appendix A is an example of what has been called "phrase linking grammar" by Peters (1980, 1981). In a phrase linking grammar, rules are interpreted as node admissibility conditions on data structures richer than the familiar trees, structures that Peters calls "linked trees". For a definition of linked trees, see Appendix A. An example of a linked tree for a topicalized sentence is shown in (2).[1]

(2) $[_S$ $[_{PP}$ le-kol yeled] $[_{S_1}$ $[_{NP}$ ani] $[_{VP}$ $[_V$ xošev] $[_{S'}$ $[_C$ še] $[_S$ $[_{NP}$ rina]
to every boy I think that Rina

$[_{VP}$ $[_V$ her'ata] $[_{NP}$ et dani] …]]]]]]
showed ACC Dani

The PP in (2) is an example of "dislocated element" (see Appendix A). The link enables the PP node to paticipate in satisfying both rules S2 and S8, repeated here as (3) and (4):

(3) $[_{VP}$ V $(XP_1 \ldots XP_n)]$
where XP_1 is NP or PP, and XP_i = PP for $1 < i \leq n$

(4) $[_S$ XP S] (Topicalization)

Since *show* is subcategorized for both an NP and a PP complement, the structure in (2) would be starred by the grammar if it didn't have the link. Figuratively speaking, the link enables the PP node to "be" at two places in the tree at the same time.

When the linked tree in (2) is interpreted by the phonological component, the link is dissolved and a phonologically null element (gap) is the realization of the missing daughter of VP. We will now see how the semantic component interprets linked trees.

First notice a general convention in my system, adopted only for the sake of simplifying the translations: all NPs and PPs that a verb is subcategorized for are translated as arguments of that verb. (No other PPs appear in the fragment.) Prepositions are therefore treated as semantically void and translations of PPs are of the same type as of NPs (see T2 and T5 in Appendix B).

A general feature of my system is stated in Appendix B as the "Translation Convention". It states that the translation X′ of every syntactic category is a triplet. The first coordinate of the triplet is called the "head" of X′ (hX′) and consists of the familiar translation into IL. The second coordinate is basically Cooper's store as proposed in Cooper (1975), which I call "quantifier store" (following Bach & Partee 1980). The third coordinate is the set of indices of the potential resumptive pronouns encountered so far in the translation, and I call it "resumptive-pronoun store". Notice that clause B of the Translation Convention ensures that only translations of the form <hS′, 0, 0> "count" for sentences, i.e. all stores must be empty at the end of the translation.

1. The indexing of nodes in trees is done purely for expository purposes and has no theoretical signifiance.

For the sentence in (2) to end up having an interpretation, the dislocated PP must be assigned the following translation:

$$<{}^{\vee}p_i, \{<\lambda P\ \forall x\ [boy'(x) \rightarrow P\{x\}], i>\}, 0>$$

p_i is the i-th variable that ranges over properties of properties of individuals. $\{\lambda P\ \forall x\ [boy'(x) \rightarrow P\{x\}], i>\}$ is the quantifier store where the familiar meaning of the NP *every boy* has been stored, together with the index of p_i. The resumptive-pronoun store in this case is 0. This translation of PP is used when translating S_1:

(5) $S_1' = <think'\ (x_0, {}^{\wedge}show'(r, d, p_i)]), \{\lambda P\ \forall x\ [boy'(x) \rightarrow P\{x\}], i>\}, 0>$

This is the same translation that the sentence would have, were the PP a "real" daughter of VP, whose meaning is stored (see the NP Storage Convention in Appendix B).[2]

We can now apply the A clause of T8 in Appendix B, which will quantify in the meaning of PP that was kept in store:[3]

(6) $$\begin{aligned} S' &= <[\lambda p_i\ think'(x_0, {}^{\wedge}show'(r, d, p_i))]\ ({}^{\wedge}\lambda P\ \forall x\ [boy'(x) \rightarrow P\{x\}]), 0, 0> \\ &= <think'\ (x_0, {}^{\wedge}show'(r, d, {}^{\wedge}\lambda P\ \forall x\ [boy'(x) \rightarrow P\{x\}])), 0, 0> \\ &= <think'\ (x_0, {}^{\wedge}\forall x\ [boy'(x) \rightarrow show'_*(r, d, x)]), 0, 0> \end{aligned}$$

The rule we have just applied "lowers" the meaning of PP into the scope of *think*. This is different from the outcome of the Store Retrieval Convention of Appendix B, that gives stored meanings scope over the whole sentence. Notice that nothing prevents us from applying this convention to (5), to get another meaning of S_1:

(7) $$\begin{aligned} S_1' &= <\lambda P\ \forall x\ [boy'(x) \rightarrow P\{x\}](\lambda x_i\ [\lambda p_i\ think'(x_0, {}^{\wedge}show'(r, d, p_i)] \\ &\quad ({}^{\wedge}PP\{x_i\})), 0, 0> \\ &= <\forall x\ [boy'(x) \rightarrow think'(x_0, {}^{\wedge}show'_*(r, d, x)], 0, 0> \end{aligned}$$

But now neither clause A nor clause B of T8 is applicable to combine PP′ with S_1', so we cannot get from this a meaning for S.

The following example shows that Hebrew allows multiple gaps.

(8) $[_S\ [_{PP}$ la-yeladim šelo$]\ [_S\ [_{NP}$ ani$]\ [_{VP}\ [_V$ batuax$]\ [_{S'}\ [_C$ še$]\ [_S$
to his kids I am-sure that

$[_{NP}$ et ha-sefer haze$]\ [_S\ [_{NP}$ dani$]\ [_{VP}\ [_V$ lo yiten …$]]]]]]]]$
ACC the-book this Dani won't give

2. Notice that there would have been differences in implicatures were PP a "real" daughter of VP rather than being topicalized as in (2). Since I shall only be interested in the truth-conditional aspects of meaning, matters of implicatures will not be represented in my translations. See Karttunen and Peters (1979) for how this could be done.

3. The notational convention I use for brackets is that brackets go around the lamda expression and its scope. I shall not write the outmot brackets in a formula, nor brackets that are immediately contained in parentheses.

The following too is grammatical:

(9) et ha-sefer haze, ani batuax še la-yeladim šelo dani lo yiten
ACC the book this I am-sure that to his kids Dani won't give

For dealing with (8) and (9), phrase linking grammars are clearly superior to Gazdar's grammars. Gazdar would have to allow at this point an infinite number of multiply slashed categories and an infinite number of derived rules, since there is no principled way to fix an upper bound on the number of gaps. (See Engdahl (1980) for an elaboration on this point). Hebrew certainly allows for three gaps and more, though of course the examples become less natural the greater the number of gaps:

(10) et ha-smartutim $haele_1$ ani lo mevin ex_2 be-mea $dolar_3$
ACC the-junk $this_1$ I don't understand how_2 for-100 $dollars_3$
mišehu hicliax $\_\__2$ limkor $\_\__1$ $\_\__3$
anybody succeeded to-sell

(11) et ha-smartutim $haele_1$, dani amar še la-šaxen $ha\text{-}kamcan_2$
ACC the-junk $this_1$ Dani said that the-neighbour $the\text{-}stingy_2$
hu lo mevin ex_3 be-mea $dollar_4$ mišehu
he doesn't understand how for-100 dollars anybody
hicliax $\_\__3$ limkor $\_\__1$ $\_\__2$ $\_\__4$
succeeded to-sell

(12) $mimi_1$ dani amar še et ha-smartutim $haele_2$ hu lo
from $whom_1$ Dani said that ACC the-junk $this_2$ he doesn't
mevin ex_3 be-mea $dollar_4$ et $rina_5$ hiclaxta $\_\__3$
understand how_3 for-100 dollars ACC Rina succeeded
lešeaxnea $\_\__5$ liknot $\_\__2$ $\_\__1$ $\_\__4$
to-convince to-buy

A Gazdar grammar revised to account for multiple gaps generates non-context-free languages just as phrase linking grammars do (both apparently generate small supersets of the context-free languages), but is in great disadvantage where the semantic interpretation is concerned. For it has no way to ensure that the right dislocated element gets quantified in for the right variable in (8) and (9). A solution to this problem suggested by Maling and Zaenen (1982) would account only for (8) and not for (9):

(13) a. $^*[_{A/D/E} \ldots B/D \ldots C/E \ldots]$
b. $[_{A/D/E} \ldots B/E \ldots C/D \ldots]$

Maling and Zaenen are simply stating in (13) that al dependencies involving gaps are nested, a generalization that (9) shows to be false. There are also examples in Norwegian and in Icelandic that falsify (13), as we shall see in Section 4.

Let us now turn to relative clauses. Hebrew relative clauses are formed with NP gaps in subject or direct object position alternating with resumptive pronouns. No preposition stranding is allowed, therefore resumptive pronouns are obligatory when relativizing on indirect object position. The examples in (14) and (15) involve gaps in subject and object positions respectively.

(14) $[_{NP2}\ [_{NP1}$ kol gever$]\ [_{S'}\ [_C$ še$]\ [_S \ldots [_{VP}\ [_V$ ohev$]\ [_{NP}$ et rina$]]]]]$
every man that loves ACC Rina

$$\begin{aligned} NP'_1 &= \langle {}^\vee p_i, \{\langle \lambda P\ \forall y\ [man'(y)\ \&\ R\{y\} \rightarrow P\{y\}], i\rangle\}, 0\rangle \\ VP' &= \langle love'({}^\wedge \lambda PP\{r\}), 0, 0\rangle \\ S' &= \langle p_i\ \{{}^\wedge love'({}^\wedge PP\{r\})\}, qsNP'_1, 0\rangle \end{aligned}$$

Since S′ and NP′ have an element in common in their quantifier stores (actually, they happen to have identical quantifier stores), the A clause of T6 can be used to get a translation of NP_2. What this rules does is first to change the NP in store: $P\ \forall y\ [man'(y)\ \&\ R\{y\} \rightarrow P\{y\}]$ into $R\ \forall y\ [man'(y)\ \&\ R\{y\} \rightarrow P\{y\}]$ (so that the property that the NP eventually combines with will replace R rather than P), and then replaces p_i in S′ by this NP.

$$\begin{aligned} NP'_2 &= \langle \lambda P[\lambda p_i p_i\{{}^\wedge love'({}^\wedge \lambda PP\{r\})\}]({}^\wedge \lambda R[\lambda P\ \forall y\ [man'(y)\ \&\ R\{y\} \rightarrow P\{y\}]](P), 0, 0\rangle \\ &= \langle \lambda P[\lambda R\ \forall y\ [man'(y)\ \&\ R\{y\} \rightarrow P\{y\}]]({}^\wedge love'({}^\wedge \lambda PP\{r\})), 0,0\rangle \\ &= \langle \lambda P\ \forall y\ [man'(y)\ \&\ love'({}^\wedge \lambda PP\{r\})(y) \rightarrow P\{y\}], 0, 0\rangle \\ &= \langle \lambda P\ \forall y\ [man'(y)\ \&\ love'_*(y, r) \rightarrow P\{y\}, 0, 0\rangle \end{aligned}$$

(15) $[_{NP2}\ [_{NP1}$ kol gever$]\ [_{S'}\ [_C$ še$]\ [_S\ [_{NP}$ rina$]\ [_{VP}\ [_V$ ohevet$] \ldots]]]]$
every man that Rina loves

$$\begin{aligned} NP'_1 &= \langle {}^\vee p_i, \{\langle \lambda P\ \forall y\ [man'(y)\ \&\ R\{y\} \rightarrow P\{y\}], i\rangle\}, 0\rangle \\ VP' &= \langle love'(p_i), qsNP'_1, 0\rangle \\ S' &= \langle \lambda PP\{r\}({}^\wedge love'(p_i)), qsNP'_1, 0\rangle \\ &= \langle love'(r, p_i), qsNP'_1, 0\rangle \end{aligned}$$

$$\begin{aligned} NP'_2 &= \langle \lambda P\ [\lambda p_i\ love'(r, p_i)]({}^\wedge \lambda R\ \lambda P\ \forall y\ [man'(y)\ \&\ R\{y\} \rightarrow P\{y\}]]\ (P)), 0, 0\rangle \\ &= \langle \lambda P\ love'(r, {}^\wedge \lambda R\ \forall y\ [man'(y)\ \&\ R\{y\} \rightarrow P\{y\}]), 0, 0\rangle \\ &= \langle \lambda P\ \forall y\ [man'(y)\ \&\ love'_*(r, y) \rightarrow P\{y\}], 0, 0\rangle \end{aligned}$$

I now give examples of relative clauses with resumptive pronouns. Parallel to (15), we have (16), where we see how the resumptive-pronoun store is used. This store is similar to the pronoun-store that Bach and Parteee (1980) argue is needed to account for anaphora. The difference is that in my system, the index of a variable used in translating a pronoun is only optionally stored. Any pronoun is potentially resumptive and the system has the option to make it a resumptive pronoun by storing the index of the variable used in its translation. This index will be used to quantify in the meaning of the head NP over the right variable, according to rules T6 or T7. Notice that the translation of a pronoun in this system has $\lambda PP\{x_i\}$ as its head, whereas the translation of a gap has ${}^\vee p_i$ as its head.

(16) $[_{NP_3}\,[_{NP_2}$ kol gever] $[_{S'}\,[_{C}$ še] $[_{S}\,[_{NP}$ rina] $[_{VP}\,[_{V}$ ohevet] $[_{NP1}$ oto]]]]]
every man that Rina loves him

$$\begin{aligned} NP'_1 &= <\lambda PP(x_i), 0, \{i\}> \\ VP' &= <love'(\hat{}\lambda PP(x_i)), 0, \{i\}> \\ S' &= <love'_*(r, x_i), 0, \{i\}> \\ NP'_2 &= <\lambda P\ \forall y\ [man'(y)\ \&\ R\{y\} \rightarrow P\{y\}], 0, 0> \end{aligned}$$

Since qsS′ is empty and rpsS′ = i, we may use the B clause of T6 to get the translation of NP_3. What this rule does is replace R in hNP_2 by $\hat{}\lambda x_i hS'$:

$$\begin{aligned} NP'_3 &= <\lambda R\ \lambda P\ \forall y\ [man'(y)\ \&\ R\{y\} \rightarrow P\{y\}](\hat{}\lambda x_i\ love'_*(r, x_i), 0, 0> \\ &= <\lambda P\ \forall y\ [man'(y)\ \&\ love'_*(r, y) \rightarrow P\{y\}], 0, 0> \end{aligned}$$

Rules S6 and S8 of Appendix A also accept the NP in (17), where the resumptive pronoun is topicalized inside the relative clause:

(17) $[_{NP3}\,[_{NP2}$ kol gever] $[_{S'}\,[_{C}$ še] $[_{S}\,[_{NP1}$ oto] $[_{S}\,[_{NP}$ rina] $[_{VP}\,[_{V}$ ohevet] …]]]]]
every man that him Rina loves

$$\begin{aligned} NP'_1 &= <{}^{\vee}p_i, \{\lambda PP(x_i\}, i>\}, (i)> \text{ (by clause B of the NP Storage Convention)} \\ VP' &= <love'(p_i), qsNP'_1, \{i\}> \\ S'_1 &= <love'(r, p_i), qsNP'_1, \{i\}> \end{aligned}$$

Since hNP_1 is ${}^{\vee}p_i$ and the quantifier stores of NP'_1 and S'_1 have an element in common, the A clause of T8 can be used to get the translation of S. What this rule does is replace p_i in hS'_1 by the store:

$$\begin{aligned} S' &= <[\lambda p_i\ love'(r, p_i)](\hat{}\lambda PP\{x_i\}), 0, \{i\}> \\ & <love'_*(r, x_i), 0, \{i\}> \end{aligned}$$

This is the same as S′ under (16). From here, we proceed as in (16) and get the same translation for NP_3.

Rule T7 accepts NP_3 in (18), where NP_2 has two sisters: NP_1 and S, rather than the familiar unique S′ sister.

(18) $[_{NP3}\,[_{NP2}$ kol gever] $[_{NP1}$ oto] $[_{S}\,[_{NP}$ rina] $[_{VP}\,[_{V}$ ohevet] …]]]
every man him Rina loves

$$\begin{aligned} NP'_2 &= <\lambda P\ \forall y\ [man'(y)\ \&\ R\{y\} \rightarrow P\{y\}], 0, 0> \\ NP'_1 &= <{}^{\vee}p_i\ \{\lambda PP\{x_i\}, i>\},\{i\}> \\ VP' &= love'\ (p_i), qsNP'_1, \{i\}> \\ S' &= <love'(r, p_i), qsNP'_1, \{i\}> \end{aligned}$$

Since hNP'_1 is ${}^{\vee}p_i$ and the quantifier stores of NP'_1 and S′ have an element in common, which is moreover $PP\{x_i\}$, we may use T7 to get the translation of NP_3. What this rule does is replace p_i in hS′ by the store and then proceed like the B clause of T6, which is the rule for relative clauses with a resumptive pronoun.

$$\begin{aligned} NP'_3 &= \langle \lambda R\, \lambda P\, \forall y\, [man'(y)\ \&\ R(y) \rightarrow P\{y\}(^\wedge\lambda x_i\, [\lambda p_i\, love'(r, p_i)] \\ &\quad (^\wedge\lambda PP\{x_i\})), 0, 0\rangle \\ &= \langle \lambda P\, \forall y\, [man'(y)\ \&\ love'_*(r, y) \rightarrow P\{y\}], 0, 0\rangle \end{aligned}$$

Notice that the way T6 and T7 are set up takes care of the fact that in (17), *oto* may be a resumptive pronoun (which in this case it is, since it happens to be the only pronoun in a relative clause with no gaps), whereas in (18), *oto* is obligatorily the resumptive pronoun (i.e. it would necessarily be the resumptive pronoun even if the clause had other pronouns). The difference can be seen in the following:

(19) a. ha-rofe še otam šalaxti elav
the-doctor that them I-sent to-him

b. *ha-rofe otam šalaxti elav
the-doctor them I-sent to-him

There are two pronouns in both (19a) and (19b). Note that *elav* agrees with the head in number, whereas *otam* does not. (19a) gets two readings by T6 that differ as to which one of the two pronouns is interpreted as a resumptive pronoun. The reading where *otam* is the resumptive pronoun gets ruled out for pragmatic reasons and (19a) ends up having one reading where *elav* is the resumptive pronoun.[4] (19b) on the other hand gets only one reading by T7 – that in which *otam* is the resumptive pronoun. This reading gets ruled out for pragmatic reasons, which results in (19c) being unacceptable.

Notice also that a structure accepted by S7 is not given a semantic interpretation unless XP is a pronoun. This rules out (20a), whereas (20b) is accepted by S6 and S8:

(20) a. *ha-iš oto ve et axiv rina ohevet
the-man him and ACC his-brother Rina loves

b. ha-iš še oto ve et axiv rina ohevet
the-man that him and ACC his-brother Rina loves

(21) is an example with a resumptive pronoun in subject position:[5]

---

4. Treating person, gender and number agreement of resumptive pronouns to the head as a pragmatic issue was suggested to me by Charles Kirkpatrick.

5. The following problem arises immediately:

(i) *kol gever še hu ohev et rina
every man that he loves ACC Rina

The generalization is that nominative resumptive pronouns may not occur in the highest S sister of COMP. The following solution has been suggested to me by Lauri Karttunen. We add

(21) kol gever še dina xoševet še hu ohev et rina
every man that Dina thinks that he loves ACC Rina

Notice that since NPs with PP heads are excluded on general grounds, we do not get PP gaps in relative clauses, only P+resumptive pronouns:[6]

(22) a. kol gever še rina xoševet alav
every man that Rina thinks about-him

b. *[$_{NP}$ [$_{PP}$ al kol gever] še rina xoševet ___]

Topicalized elements may, on the other hand, be PPs (by S8), so that both (23a) and (23b) are acceptable:

(23) a. kol gever, rina xoševet alav
every man Rina thinks about-him

b. al kol gever rina xoševet
about every man Rina thinks

---

a new pronoun store called "local resumptive pronoun store", in which we store the indices of the variables translating nominative pronouns. The indices for all the other pronouns are stored as before in the resumptive pronoun store. At the stage where we combine the interpretation of S′ with the interpretation of its sister node, whatever it may be, we transfer the contents of the local resumptive pronoun store into the pronoun store. If that sister node happened to be the head NP, we would have already retrieved an index from the resumptive pronoun store and this index could not be one for a nominative pronoun in the highest S.

6. Definite NPs in object position are marked in Hebrew by the Acc marker *et*. This is not the case in (15), repeated here as (ii), the acceptable counterpart of (22b):

(ii) [$_{NP}$ kol gever] še rina ohevet ___
every man that Rina loves

The reason is that the case marking of the whole NP percolates to the head NP. For example:

(iii) kol gever še rina ohevet ohev ota
every man that Rina loves loves her

In (iii), *kol gever* is nominative since the NP *kol gever še rina ohevet* is the subject of the sentence. The rule of Acc marking would apply therefore only to NPs that are not directly dominated by NP.

I still have to explain why there is no preposition stranding in Hebrew, i.e. why (iv) is unacceptable where (22a) was acceptable:

(iv) *[kol gever] še rina xoševet al ___
every man that Rina thinks about

The reason, I think, has to do with the fact that prepositions in Hebrew are viewed as case-markings on NPs, and therefore have to be adjacent to those NPs.

Since dislocated PPs are necessarily link children (see the specification of dislocated constituents in Appendix A), the following is ungrammatical:

(24) *al kol gever rina xoševet alav
about every man Rina thinks about-him

Only dislocated NPs that are link children can be marked with the Acc marker, since only NP sisters of V get marked Acc. Since linking is not used in accounting for resumptive pronouns, it follows that (25a) is ungrammatical and (25b) is good.

(25) a. *et dani rina ohevet oto
ACC Dani Rina loves him

b. [et dani] rina ohevet ___

Finally, note that examples such as (26a) have nothing to do with topicalization and are quite distinct from those like (23b). (26a) is an example of the Hebrew subject-verb inversion rule, that is optionally triggered by fronting an element of the verb's complement structure. This rule is not at all the same as topicalization, as it is not unbounded (cf. 26b).

(26) a. al kol gever xoševet rina
about every man thinks Rina

b. *al kol gever amar dani še rina xoševet ___
about every man said Dani that Rina thinks

Another difference is that Topicalization may involve a resumptive pronoun (cf. 23a), whereas the subject-verb inversion rule does not involve a pronoun to replace the fronted element (which is as expected, since *think* is not subcategorized for two *about* complements):

(27) *kol gever xoševet rina alav
every man thinks Rina about-him

In summary, Hebrew has a rule of Topicalization (S8), where we find NP or PP preceding S and where either the "linking" strategy is used or the resumptive pronoun strategy. Relativization, on the other hand, involves an NP preceding S′ and again either strategy may be used (cf. S6). Additionally, there is the tripartite NP construction for relativization (cf. S7), where an NP is followed first by a resumptive pronoun and then by S.

3. Differences between resumptive pronouns and gaps

3.1 Syntactic differences

Our system still needs a constraint to block examples such as the following, similar to what is the case in English:

(28) *[$_{NP}$ [$_{NP}$ ha-iš] [$_{S'}$ [$_{C}$ še] [$_{S}$ [$_{VP}$ [$_{V}$ raiti] [et [$_{NP}$ dani ve …]]]]]]
the man that I-saw ACC Dani and

Assume the constraint is stated as follows:

(29) If X directly dominates [X CONJ X … CONJ X], then every link descendant of the root X is also a link descendant of each daughter X.

If we treat resumptive pronouns as phonological realizations of gaps, we get the following counterexample to (29):

(30) ha-iš še raiti et dani ve oto
the-man that I-saw ACC Dani and him

whereas, if resumptive pronouns are independent nodes, (29) holds with no problems. A similar example was given in (1), repeated here as (31). (31) is acceptable not only in Hebrew, but also in Irish and in the Scandinavian languages.

(31) This is the woman that John said that she and Bill are having an affair

Another example that conforms to (29) is brought out in (32):

(32) [$_{NP}$ [$_{NP}$ ha-iš] [$_{S'}$ [$_{C}$ še] [$_{S}$ [$_{S}$ [$_{NP}$ dani] [$_{VP}$ [$_{V}$ sone] …]] [$_{CONJ}$ ve]
the-man that Dani hates and
[$_{S}$ [$_{NP}$ rina] [$_{VP}$ [$_{V}$ ohevet] …]]]]]
Rina loves

If resumptive pronouns were realizations of gaps, the following should be just as grammatical as (32), which it is not. And indeed, under my analysis, (29) explains its ungrammaticality:

(33) *[$_{NP}$ [$_{NP}$ ha-iš] [$_{S'}$ [$_{C}$ še] [$_{S}$ [$_{S}$ [$_{NP}$ dani] [$_{VP}$ [$_{V}$ sone] [$_{NP}$ oto]]] [$_{CONJ}$ ve]
the-man that Dani hates him and
[$_{S}$ [$_{NP}$ rina][$_{VP}$ [$_{V}$ ohevet] …]]]]]
Rina loves

whereas if *oto* is seen just as the phonological realization of another link that starts at the first VP and ends at the head, the ungrammaticality of (33) is unexplained.

So I have established a syntactic distinction between gaps and resumptive pronouns. We now turn to semantic distinctions.

3.2 Semantic differences

3.2.1 *Relative clauses with both gaps and resumptive pronouns*

I will now show that, without any additional stipulation, we get the right result when a relative clause contains both a gap and a pronoun. In this case, the pronoun is never interpreted as a resumptive pronoun, rather it is the gap that gets bound by the head:[7]

(34)	ha-iša	še	dani	her'a	la	___
	the-woman$_i$	that	Dani	showed	to-her	[__]$_i$
	"the woman that Dani showed to her"					

whereas, in the case of two pronouns, either could be bound by the head:

(35)	ha-iša	še	dani	her'a	la	ota
	the-woman	that	Dani	showed	to-her	her
	"the woman that Dani showed to her" (same meaning as 34)					
	or "the woman to whom Dani showed her"					

The representation for (34) is (36):[8]

(36) $[_{NP3}\ [_{NP2}$ ha-iša$]\ [_{S'}\ [_{C}$ še$]\ [_{S}\ [_{NP1}$ dani$]\ [_{VP}\ [_{V}$ her'a$]\ [_{PP}$ la$]$ …$]]]]$
the-woman that Dani showed to-her

$$
\begin{aligned}
NP'_1 &= <\lambda PP\{d\}, 0, 0> \\
PP' &= <\lambda PP\{x_j\}, 0, \{j\}> \\
NP'_2 &= <{}^{\vee}p_i, \{<\lambda P\ \exists y\ [\forall z\ [(woman'(z)\ \&\ R\{z\} \leftrightarrow z = y]\ \&\ P\{y\}, i>\}\ 0> \\
VP' &= <show'(p_i, {}^{\wedge}\lambda PP\{x_j\}), qsNP'_2, \{j\}> \\
S' &= <show'(d, p_i, {}^{\wedge}\lambda PP\{x_j\}, qsNP'_2, \{j\}> \\
NP'_3 &= <\lambda P\ [\lambda p_i\ show'(d, p_i, {}^{\wedge}\lambda PP\{x_j\})]\ ({}^{\wedge}\lambda RqsNP'_2\ (P)), 0, \{j\}> \\
&= <\lambda P\ \exists! y\ [woman'(y)\ \&\ show'_*(d, y, x_j)\ \&\ P\{y\}], 0, \{j\}>
\end{aligned}
$$

Notice that x_j cannot be bound by NP′ since, when qsS′ is not empty, it is the variable whose index is stored in qsS′ that gets bound, in this case p_i. x_j may be bound by a head

7. Notice that pronominal PPs in Hebrew precede NPs (even pronominal NPs) in the VP. For example:

(iv)	a.	natati	lo	oto
		I-gave	to-him	it
	b.	?natati oto lo		

8. The reader is reminded that I use relational notation, e.g. A (B, C), not only when these denote expressions of type t, but also when they denote expressions of type ⟨e, t⟩ (cf. T2). Therefore in show′(p_i, PP{x_j}) below, p_i is the direct object and not the subject. Notice moreover that in the translations under (36), and everywhere else in the paper, I use qsX′ ambiguously to refer to the quantifier store ⟨α, i⟩ and also to its first coordinate α. It should be clear each time which one is intended.

further up the tree or by another dislocated element. Notice that I am for simplicity writing down only one possible translation of the pronoun *la*. The other one is simply $<\lambda PP\{x_j\}, 0, 0>$, i.e. the meaning of a regular pronoun rather than that of a resumptive pronoun.

The representation for (35) is (37):

(37) $[_{NP3}$ $[_{NP}$ ha-iša] $[_{S'}$ $[_{C}$ še] $[_{S}$ $[_{NP2}$ dani] $[_{VP}$ $[_{V}$ her'a] $[_{PP}$ la] $[_{NP1}$ ota]]]]]
the woman that Dani showed to-her her

$$NP'_1 = <\lambda PP\{x_i\}, 0, \{i\}>$$
$$PP' = <\lambda PP\{x_j\}, 0, \{j\}>$$
$$NP'_2 = <\lambda PP\{d\}, 0, 0>$$
$$S' = <show'(d, x_i, x_j), 0, \{i, j\}>$$

The semantics will give us the right ambiguity, since depending on which index is retrieved from rpsS′, the meanings of NP_3 will be:

$$<[\lambda R\ \lambda P\ \exists!y\ [woman'(y)\ \&\ R\{y\}\ \&\ P\{y\}]]\ (^\wedge\lambda x_i\ show'_*(d, x_i, x_j)), 0, \{j\}>$$

or

$$<[\lambda R\ \lambda P\ \exists!y\ [woman'(y)\ \&\ R(y)\ \&\ P\{y\}]]\ (^\wedge\lambda x_j\ show'_*(d, x_i, x_j)), 0, \{i\}>$$

and, after lamda conversion:

$$<\lambda P\ \exists!y\ [woman'(y)\ \&\ show'_*(d, y, x_j)\ \&\ P\{y\}], 0, \{i\}>\ \text{(same as for 36)}$$

or

$$<\lambda P\ \exists!y\ [woman'(y)\ \&\ show'_*(d, x_i, y)\ \&\ P\{y\}], 0, \{i\}>$$

3.2.2 *Coindexing of gaps and resumptive pronouns*

Consider the following examples:

(38) a. ha-iš še im-o ohevet ___
the-man that mother-his loves

b. ha-iš še im-o ohevet oto
the-man that mother-his loves him

(38b) uses a resumptive pronoun where (38a) has a gap. Even though this is the only difference between them, the two NPs do not have the same readings. In (39) and (40), we present all the coindexing possibilities for (38a) and (38b) respectively:[9]

(39) the man_1 that his_2 mother loves $\_\_\__1$

9. I use the term "coindexed" (rather than "coreferential") in the sense emphasized by Bach and Partee (1980): " … coindexing a pronoun with some other expression is a shorthand of saying that the pronoun in question is being interpreted as a bound-variable …" (p. 7).

(40) a. the man_1 that his_2 mother loves him_1
b. the man_1 that his_1 mother loves him_1
c. the man_1 that his_1 mother loves him_2

Notice that (38a) has only one reading, it does not have a reading where the pronoun and the gap are coindexed. The gap is of course always coindexed with the head, therefore the head and the pronoun are not coindexed. In other words, (38a) does not have a reading where the pronoun is a resumptive pronoun for *the man*. (38b), on the other hand, has a reading where the two pronouns are coindexed, i.e. (40b). When they are not, either can be the resumptive pronoun. This is shown in (40a) and (40c).

It is interesting now to see that our system gives exactly the right readings for (38a) and (38b). We will see that (38a) gets the meaning in (41), that corresponds to (39):[10]

(41) $\lambda P\, \exists y\, [\text{mother}'(y)\ \&\ \text{possess}'_*(z, y)\ \&\ \exists! x\, [\text{man}'(x)\ \&\ \text{love}'_*(y, x)\ \&\ P\{x\}]]$

whereas (38b) gets the meanings in (42a), (42b), (42c) corresponding to the readings in (40a), (40b), (40c):

(42) a. $\lambda P\, \exists! x\, [\text{man}'(x)\ \&\ \exists y\, [\text{mother}'(y)\ \&\ \text{possess}'_*(z, y)\ \&\ \text{love}'_*(y, x)]\ \&\ P\{x\}]$
b. $\lambda P\, \exists! x\, [\text{man}'(x)\ \&\ \exists y\, [\text{mother}'(y)\ \&\ \text{possess}'_*(x, y)\ \&\ \text{love}'_*(x, y)]\ \&\ P\{x\}]$
c. $\lambda P\, \exists! x\, [\text{man}'(x)\ \&\ \exists y\, [\text{mother}'(y)\ \&\ \text{possess}'_*(x, y)\ \&\ \text{love}'_*(y, z)]\ \&\ P\{x\}]$

The crucial point is the following: the variable *z* in (41), that stands for the pronoun *his* in (39), is outside the scope of the head NP, which is the quantifier that binds the variable *x* (that stands for the gap in (39)). Therefore, even if, while translating (38a), we assign the pronoun and the gap translations the same index for the respective variables, eventually one will be bound and the other not. Notice that (41) can really be rewritten as (43):

(43) $\lambda P\, \exists y\, [\text{mother}'(y)\ \&\ \text{possess}'_*(x, y)\ \&\ \exists! x\, [\text{man}'(x)\ \&\ \text{love}'_*\, (y, x)\ \&\ P\{x\}]]$

As pointed out to me by Charles Kirkpatrick, I still have to show why we do not get accidental binding in (38a). The answer, I believe, lies in the domain of pragmatics, as argued by Reinhart (1978, 1981b) for a similar question. Since the language has

10. I do not claim this is the best possible translation for *his mother*, but it will do for the purposes of this paper. Also, I will use a (somewhat misleading) notation, according to which the translation of *the woman*, for example, looks like (i), but means (ii):

(i) $\lambda P\, \exists! y[\text{woman}'(y)\ \&\ R\{y\}\ \&\ P\{y\}]$

(ii) $\lambda P\, \exists y\, [\forall z\, [\text{woman}'(z)\ \&\ R\{z\}) \leftrightarrow z = y]\ \&P\{y\}]$

the means to indicate that it intends the head NP to bind the pronoun (i.e. by using another pronoun in place of the gap), it would be infelicitous of the speaker to use (38a), when he intends to communicate (40b).

In (42), the situation is different. *x* stands for the resumptive pronoun (cf. the difference between 42a and 42c). But if the other pronoun is translated using the same variable as in the translation of the resumptive pronoun, resulting for example in *x* in (42b) where there is *z* in (42a), this occurrence of the variable will be bound by the quantifier that binds the other occurrences of *x*. Therefore, we do get in (38b) a reading where the two pronouns are coindexed.[11]

We still have to show how our system gives the right meanings. I will only show how to get the translations of (38a) and (38b), where we do choose the variables with the same index twice in the translations, since this is the interesting case.

Under (44), I show the relevant translation of (38a).

(44) $[_{NP3}\, [_{NP2}$ ha-iš$]\; [_{S'}\; [_{C}$ še$]\; [_{S}\; [_{NP1}$ imo$]\; [_{VP}\; [_{V}$ ohevet$]$ …$]]]]$
the-man that mother-his loves

$$\begin{aligned}
NP'_1 &= \langle \lambda P\, \exists y\, [mother'(y)\; \&\; possess'_*(x, y)\; \&\; P\{y\}], 0, 0\rangle \\
NP'_2 &= \langle {}^{\vee}p_i, \{\lambda P\, \exists! x\, [man'(x)\; \&\; R\{x\}\; \&\; P\{x\}], i\rangle\}, 0\rangle \\
S' &= hNP'_1({}^{\wedge}love'(p_i)), qsNP'_2, \{i\}\rangle \\
&= \langle \exists y\, [mother'(y)\; \&\; possess'_*(x, y)\; \&\; love'(y, p_i)], qsNP'_2, 0\rangle \\
NP'_3 &= \langle \lambda P\, [p_i hS']\, ({}^{\wedge}\lambda R qsNP'_2\, (P)), 0, 0\rangle \\
&= \langle \lambda P\, [\lambda p_i\, [\exists y\, [mother'(y)\; \&\; possess'_*(x, y) \\
&\quad \&\; love'(y, p_i)]\, ({}^{\wedge}\lambda R qsNP'_2\, (P)), 0, 0\rangle \\
&= \langle \lambda P\, \exists y\, [mother'(y)\; \&\; possess'_*(x, y) \\
&\quad \&\; love'(y, {}^{\wedge}\lambda R\, \exists! x\, [man'(x)\; \&\; R(x)\; \&\; P(x)]], 0, 0\rangle \\
&= \langle \lambda P\, \exists y\, [mother'(y)\; \&\; possess'_*(x, y)\; \&\; \exists! x\, [man'(x) \\
&\quad \&\; love_*(y, x)\; \&\; P\{x\}]], 0, 0\rangle
\end{aligned}$$

hNP'_3 is indeed the reading in (43). (Notice that nothing would be changed had we stored anything in the resumptive pronoun stores.)

---

11. The general question of where it is permissible to use the same variable in the translation of two pronouns is beyond the scope of this paper. See Keenan (1974), Reinhart (1979, 1981b) or Bach and Partee (1980) for different approaches to the question of anaphora. I will assume that we use the same variable for both pronouns in order to get the readings in (va) and (vb).

(v) a. imo ohevet oto
mother-his$_1$ loves him$_1$

b. imo ohevet et dani
mother-his$_1$ loves ACC Dani$_1$

Under (45), I give the translation of (38b).

(45) [$_{NP4}$ [$_{NP3}$ ha-iš] [$_{S'}$ [$_C$ še] [$_S$ [$_{NP_2}$ im-o] [$_{VP}$ [$_V$ ohevet] [$_{NP}$ oto]]]]]
the-man that mother-his loves him

$$
\begin{aligned}
NP'_1 &= \langle \lambda P P\{x_i\}, 0, \{i\} \rangle \\
NP'_2 &= \langle \lambda P\, \exists y\, [\text{mother}'(y)\ \&\ \text{possess}'_*(x_i, y)\ \&\ P\{y\}], 0, 0 \rangle \\
S' &= \langle \exists y\, [\text{mother}'(y)\ \&\ \text{possess}'_*(x_i, y)\ \&\ \text{love}'_*(y, x_i)], 0, \{i\} \rangle \\
NP'_3 &= \langle \lambda P\, \exists! x\, [\text{man}'(x)\ \&\ R\{x\}\ \&\ P\{x\}], 0, 0 \rangle \\
NP_4 &= \langle \lambda P\, \exists! x\, [\text{man}'(x)\ \&\ \exists y\, [\text{mother}'(y)\ \&\ \text{possess}'_*(x, y) \\
&\qquad \&\ \text{love}'_*(y, x)]\ \&\ P\{x\}]\ 0, 0 \rangle
\end{aligned}
$$

hNP'_4 is indeed the translation in (42b).

Notice that we could not have explained (38a) by a general prohibition on coindexing gaps and pronouns, since the following is acceptable:

(46) ha-iš še ___ ohev et im-o
the-man$_1$ that ___$_1$ loves ACC moher-his$_1$

My system gets this reading:

(47) [$_{NP3}$ [$_{NP2}$ ha-iš] [$_{S'}$ [$_C$ še] [$_S$ … [$_{VP}$ [$_V$ ohev][$_{NP1}$ et im-o]]]]]
the man that loves ACC mother-his

$$
\begin{aligned}
NP'_1 &= \langle \lambda P\, \exists y\, [\text{mother}'(y)\ {}^\wedge\text{possess}'_*(x, y)\ \&\ P\{y\}], 0, 0 \rangle \\
NP'_2 &= \langle {}^\vee p_i, \{\langle \lambda P\, \exists! x\, [\text{man}'(x)\ \&\ R\{x\}\ \&\ P\{x\}], i \rangle\}, 0 \rangle \\
S' &= \langle p_i\, ({}^\wedge\text{love}'({}^\wedge hNP'_1)\}, qsNP'_2, 0 \rangle \\
NP_3 &= \langle \lambda P\, [\lambda p_i p_i\{{}^\wedge\text{love}'({}^\wedge hNP'_1)\}]\, ({}^\wedge\lambda R qsNP'_2\, (P)), 0, 0 \rangle \\
&= \langle \lambda P\, [\lambda R\, \exists! x[\text{man}'(x)\ \&\ R\{x\}\ \&\ P\{x\}]]\, ({}^\wedge\text{love}({}^\wedge hNP'_1)), 0, 0 \rangle \\
&= \langle \lambda P\, \exists! x[\text{man}'(x)\ \&\ \text{love}'(x, {}^\wedge hNP'_1)\ \&\ P\{x\}], 0, 0 \rangle \\
&= \langle \lambda P\, \exists! x\, [\text{man}'(x)\ \&\ \exists y\, [\text{mother}'(y)\ \&\ \text{possess}'_*(x, y) \\
&\qquad \&\ \text{love}'_*(x, y)], 0, 0 \rangle
\end{aligned}
$$

hNP'_3 is the reading in (46).

3.2.3 *Referentiality of the head of the relative clause*

I will now show that the ways in which the binding of gaps differs from the binding of resumptive pronouns gives us the right scope results. Consider the following NP:

(48) [$_{NP3}$ [$_{NP2}$ ha-iša] [$_{S'}$ [$_C$ še] [$_S$ [$_{NP_1}$ hu] [$_{VP}$ [$_V$ mexapes] …]]]]
the-woman that he seeks

$$
\begin{aligned}
NP'_1 &= \langle \lambda P P\{x\}, 0, 0 \rangle \\
NP'_2 &= \langle {}^\vee p_i, \{\langle \lambda P\, \exists! y\, [\text{woman}'(y)\ \&\ R\{y\}\ \&\ P\{y\}]\ i \rangle\}, 0 \rangle \\
VP' &= \langle \text{seek}'(p_i), qsNP'_2, 0 \rangle \\
NP'_3 &= \langle \lambda P\, [\lambda p_i\ \text{seek}'(x, p_i)]({}^\wedge\lambda R qsNP'_2\, (P)), 0, 0 \rangle \\
&= \langle \lambda P\ \text{seek}(x, {}^\wedge R\, \exists! y\, [\text{woman}'(y)\ \&\ R\{y\}\ \&P\{y\}]), 0, 0 \rangle
\end{aligned}
$$

And indeed the following sentence has a *de dicto* reading:

(49)	dani	yimca	et	ha-iša	še	hu	mexapes
	Dani	will-find	ACC	the woman	that	he	seeks

which is the following (where *dani* binds x):[12]

$$seek'(d, {}^\wedge\lambda R\, \exists y\, [woman'(y)\ \&\ willfind'(d, y)\ \&\ R\{y\}])$$

The interesting point is that the sentence parallel to (49), but where the relative clause is formed with a resumptive pronoun, does not have a *de dicto* reading:

(50)	dani	yimca	et	ha-iša	še	hu	mexapes	ota
	Dani	will-find	ACC	the-woman	that	he	seeks	her

The only meaning of this sentence can be paraphrased as follows: "there is a woman that Dani is seeking and he will find this woman". And indeed, under my account, the NP that contains the relative clause has the following structure:

(51) $[_{NP4}\ [_{NP3}$ ha-iša$]\ \ [_{S'}\ [_C$ še$]\ [_S\ [_{NP_1}$ hu$]\ [_{VP}\ [_V$ mexapes$]\ [_{NP1}$ ota$]]]]]$
the woman that he seeks her

$$\begin{aligned}
NP'_1 &= \langle \lambda P P\{x_i\}, 0, \{i\}\rangle \\
VP' &= \langle seek'(\lambda P P\{x_i\}, 0, \{i\}\rangle \\
NP'_2 &= \langle \lambda P P\{x_j\}, 0, 0\rangle \\
S' &= \langle seek'(x_j, \lambda P P\{x_i\}, 0, \{i\}\rangle \\
NP_3 &= \langle \lambda P\, \exists! y\, [woman'(y)\ \&\ R\{y\}\ \&\ P\{y\}], 0, 0\rangle \\
NP'_4 &= \langle [\lambda R h NP'_3]({}^\wedge\lambda x_i\ seek'(x_j, \lambda P P\{x_i\})), 0, 0\rangle \\
&= \langle \lambda P\, \exists! y\, [woman'(y)\ \&\ seek'(x_j, \lambda P P\{y\})\ \&\ P\{y\}], 0, 0\rangle
\end{aligned}$$

Therefore in the case where x_j gets bound by *dani′*, the only meaning for (50) is the following:

$$\exists! y\, [woman'(y)\ \&\ seek'(d, \lambda P P\{y\})\ \&\ willfind'(d, y)]$$

i.e. the only reading we get for (50) is the *de re*, which is the right result.

3.2.4 *Island constraints*

Next, we turn to the difference between relative clauses with gaps and resumptive pronouns with respect to gaps bound from outside the clause.

Consider the following grammatical sentence of Hebrew:

(52) $[_S\ [_{PP}$ la-yeled haze$]\ [_S\ [_{Adv}$ od lo$]\ [_{VP}\ [_V$ macati$]\ [_{NP}\ [_{NP}$ sefer$]$
to-this kid yet not I-found a-book

$[_{S'}\ [_C$ še$]\ [_S\ [_{VP}\ [_{Adj}$ keday $]\ [_{VP}\ [_V$ latet $]\ \ldots]]]]]]]]$
that it-is-worth to-give

12. A treatment for tense is outside the scope of this paper. I use *will-find* here rather than *find* so that the reading does not sound contradictory.

Surprisingly, the corresponding sentence with a resumptive pronoun in the relative clause, i.e. with only one gap instead of the two in (52), is ungrammatical:

(53) $[_S [_{PP}$ la-yeled haze$]$ $[_S [_{Adv}$ od lo$]$ $[_{VP} [_V$ macati$]$ $[_{NP} [_{NP}$ sefer$]$
to this kid yet not I-found a book
$[_{S'} [_C$ še$]$ $[_S [_{VP} [_{Adj}$ keday $]$ $[_{VP} [_V$ latet$]$ $[$oto$]]]]]]]]]$
that it-is-worth to-give it

Yet, the constraint that will star (53) cannot be syntactic, since the same phenomenon repeats itself when, instead of a constituent dislocated from within the relative clause, we have an NP that syntactically is inside the relative clause, but semantically is "quantified" into that clause. (54) has a reading where the pronoun *lo* is bound by *kol gever*, whereas (55) does not have such a reading:

(54) ha-iša$_1$ še kol gever$_2$ baxar $\_\__1$ tišlax lo$_2$ tmuna
the woman$_1$ that every man$_2$ chose $\_\__1$ will-send him$_2$ a-picture

(55) *ha-iša$_1$ še kol gever$_2$ baxar ota$_1$ tišlax lo$_2$ tmuna
the woman$_1$ that every man$_2$ chose her$_1$ will-send him$_2$ a-picture

The constraint that accounts for the unacceptability of both (53) and (55) will therefore be semantic. In my system, (53) does not get any reading and (55) does not get a reading where *kol gever* binds *lo*. To exemplify how this works, we now show how we get the reading in (54) and how we don't get the reading in (55).

(56) $[_{S2} [_{NP4} [_{NP1}$ ha-iša $]$ $[_{S'}[_C$ še$]$ $[_{S1} [_{NP_2}$ kol gever$]$ $[_{VP} [_V$ baxar$]$ …$]]]]]$
the woman that every man chose
$[_{VP2} [_V$ tišlax$]$ $[_{PP}$ lo$]$ $[_{NP3}$ tmuna$]]]$
will-send him a-picture

$$
\begin{aligned}
NP'_1 &= \langle {}^{\vee}p_i, \{\langle \lambda P\ \exists! y\ [woman'(y)\ \&\ R\{y\}\ \&\ P\{y\}], i\rangle\}\ 0\rangle \\
NP'_2 &= \langle {}^{\vee}p_j\ \{\lambda P\ \forall x\ [man'\{x\} \rightarrow P\{x\},]\ j\rangle, 0\rangle \\
VP'_1 &= \langle choose'(p_i), qsNP'_1, 0\rangle \\
S'_1 &= \langle p_j\ \{{}^{\wedge}choose'(p_i)\}, qsNP'_1 \cup qsNP'_2, 0\rangle \\
NP'_4 &= \langle \lambda P\ [\lambda p_i hS'_1]\ ({}^{\wedge}RqsNP'_1(P)), qsNP'_2, 0\rangle \\
&= \langle \lambda P\ [\lambda p_i p_j\ \{{}^{\wedge}choose'(p_i)\}]\ ({}^{\wedge}\lambda R\ [\lambda P\ \exists y\ [woman'(y)\ \&\ R\{y\} \\
&\qquad \&\ P\{y\}]]\ (P)), qsNP'_2, 0\rangle \\
&= \langle \lambda P\ p_j\{{}^{\wedge}choose'({}^{\wedge}\lambda R \exists! y\ [woman'(y)\ \&\ R\{y\}\ \&\ P\{y\}])\}, qsNP'_2, 0\rangle \\
NP'_3 &= \langle \lambda P\ \exists z[picture'(z)\ \&\ P\{z\}], 0, 0\rangle \\
PP' &= \langle \lambda PP\{x_j\}, 0, 0\rangle \\
VP'_2 &= \langle send'({}^{\wedge}hNP'_3, {}^{\wedge}\lambda PP\{x_j\}), 0, 0\rangle \\
S'_2 &= \langle qsNP'_2\ ({}^{\wedge}\lambda x_j\ [\lambda p_j hNP'_4({}^{\wedge}send'({}^{\wedge}hNP'_3, {}^{\wedge}\lambda PP\{x_j\}))]\ ({}^{\wedge}\lambda PP\{x_j\})), 0, 0\rangle \\
&= \langle qsNP'_2\ ({}^{\wedge}\lambda x_j\ \exists! y\ [woman'(y)\ \&\ choose'_*(x_j, y) \\
&\qquad \&\ send'(y, {}^{\wedge}hNP'_3, \lambda PP\{x_j\}]), 0, 0\rangle \\
&= \langle qsNP'_2\ ({}^{\wedge}x_j\ \exists! y\ [woman'(y)\ \&\ choose'_*(x_j, y) \\
&\qquad \&\ \exists z\ [picture'(z)\ \&\ send'_*(y, z, x_j)]]), 0, 0\rangle \\
&= \langle \forall x\ [man'(x) \rightarrow \exists! y\ [woman'(y)\ \&\ choose'_*(x, y) \\
&\qquad \&\ \exists z\ [picture'(z)\ \&\ send'_*(y, z, x)]]], 0, 0\rangle
\end{aligned}
$$

This indeed is the reading where *kol gever* has scope over the whole sentence. To show that (55) does not have this reading, we give its structure in (57):

(57) $[_{S2}\ [_{NP4}\ [_{NP3}$ ha-iša] $[_{S'}[_{C}$ še] $[_{S1}\ [_{NP2}$ kol gever] $[_{VP}\ [_{V}$ baxar] $[_{NP1}$ ota]]]]]]
the woman that every man chose her

$[_{VP2}\ [_{V}$ tišlax] $[_{PP}$ lo] $[_{NP}$ tmuna]]]
will-send him a-picture

$$\begin{aligned}
NP'_1 &= \langle \lambda P P\{x_i\}, 0, \{i\}\rangle \\
NP'_2 &= \langle {}^{\vee}p_j, \{\langle \lambda P\ \forall x\, [man'(x) \rightarrow P\{x\}], j\rangle\}, 0\rangle \\
S'_1 &= \langle p_j\ \{choose'({}^{\wedge}\lambda P P\{x_i\})\}, qsNP'_2, \{i\}\rangle \\
NP'_3 &= \langle \lambda P\, \exists! y\, [woman'(y)\ \&\ R\{y\}\ \&\ P\{y\}], 0, 0\rangle
\end{aligned}$$

To combine NP'_3 with S'_1 to get NP_4, we cannot apply A in the definition of T6 since $qsNP_3$ is empty and we cannot apply B since qsS_1 is not empty. Therefore S_2 in (57) does not get a meaning where *kol gever* has scope over the whole sentence.

The same difference shows up between topicalization with and without a resumptive pronoun (cf. rule S7 in Appendix A):

(58) a. et im-o, kol gever ohev
ACC mother-his$_1$ every man$_1$ loves

b. *im-o, kol gever ohev ota
mother-his$_1$ every man$_1$ loves her

The structures for (58a) and (58b) are shown in (59a) and (59b), respectively:

(59) a. $[_{S2}\ [_{NP2}$ et imo] $[_{S1}\ [_{NP1}$ kol gever] $[_{VP}\ [_{V}$ ohev] …]]]
ACC mother-his every man loves

$$\begin{aligned}
NP'_1 &= \langle {}^{\vee}p_i, \{\langle \lambda P\ \forall x[man'(x) \rightarrow P\{x\}], i\rangle\}, 0\rangle \\
NP'_2 &= \langle {}^{\vee}p_j, \{\langle \lambda P\ \exists y\, [mother'(y)\ \&\ possess'_*(x_i, y)\ \&\ P\{y\}], j\rangle\}, 0\rangle \\
S'_1 &= \langle p_i\ \{love'(p_j)\}, qsNP'_i \cup qsNP'_2, 0\rangle
\end{aligned}$$

b. $[_{S4}\ [_{NP4}$ imo] $[_{S3}\ [_{NP1}$ kol gever] $[_{VP}\ [_{V}$ ohev] $[_{NP3}$ ota]]]]
mother-his every man loves her

$$\begin{aligned}
NP'_1 &= \langle {}^{\vee}p_i, \{\langle \lambda P\ \forall x[man'(x) \rightarrow P\{x\}], i\rangle\}, 0\rangle \\
NP'_4 &= \langle \lambda P\ \exists y\, [mother'(y)\ \&\ possess'_*(x_i, y))\ \&\ P\{y\}], 0, 0\rangle \\
NP'_3 &= \langle \lambda P P\{x_j\}, 0, \{j\}\rangle \\
S'_3 &= \langle p_i\ \{love'(p_j)\}, qsNP'_1, \{j\}\rangle
\end{aligned}$$

Notice that we should be allowed to use the same variable x_i both in the translation of NP_1 and NP_2 (and NP_4), since we ould have to do the same to get the following reading of (60):

(60) kol gever ohev et imo
every man$_1$ loves ACC mother-his$_1$

Any element in qsS'_1 can be retrieved at this point. If the first one is, *kol gever* won't have wide scope over *imo*. If the second one is, we won't be able to combine NP_2 with S_1, because we will be missing the right element in store. So to get the reading we want, no element is retrieved from store at this point and the translation for S_2 is:

$$S'_2 \quad = \quad \langle [\lambda p_j hS'_1]\,(qsNP'_2),\ qsNP'_1,\ 0 \rangle$$

After retrieving $qsNP'_1$, I get the reading we wanted for (58a):

$$hS'_2 \quad = \quad \forall x\,[man'(x) \rightarrow \exists y\,[mother'(y)\ \&\ possess'(x, y)\ \&\ love'(x, y)]$$

S'_3, on the other hand, cannot be combined to the dislocated element NP'_4. Clause A of T8 does not apply, since $qsS'_3 \cap qsNP'_4 = 0$. Clause B of T8 does not apply twice, since $qsS'_3 \neq 0$. So we cannot get a reading for S_4 where *kol gever* has wide scope over *imo*. The only meaning we get for S_4 is when we store nothing for NP_1, and that reading would be:

$$\exists y\,[mother'(y)\ \&\ possess'(x, y)\ \&\ \forall x[man'(x) \rightarrow love'\,(x, y)]]$$

(i.e. where *imo* is outside the scope of *kol gever*.)

To summarize Section 3.2, I have shown several differences in the meanings of relative clause with and without resumptive pronouns. These differences have to do with the fact that the antecedent of a resumptive pronoun always has wider scope than any other quantifier in the same clause with the pronoun and than the antecedent of any gap in the same clause with the pronoun. The same differences appeared in the meanings of sentences topicalized with and without resumptive pronouns. My system captures these differences by ensuring that pronouns are not treated as resumptive as long as there still is unretrieved quantifier storage, i.e. as long as there still are gaps in the clause that have not been bound or NP meanings that have not been quantified in. Treatments that conflate gaps and resumptive pronouns would be hard pressed to account for these differences.

4. The distribution of resumptive pronouns

I now turn to show how my system captures the patterns of gaps and resumptive pronouns distribution in multiple extractions noted by Engdahl (1980) and Maling and Zaenen (1982). The same patterns basically hold for Hebrew, so I will start with Hebrew examples:

(61) a. ha-ma'amarim haele$_1$, dani xošev še et ha-orex
these articles$_i$ Dani thinks that ACC the editor
ha-xadaš$_2$ efšar lešaxnea ___$_2$ levater alehem$_1$
the new$_2$ it-is-possible to-convince ___$_2$ to-give-up on-them$_1$

b. *al ha-ma'amarim haele$_1$, dani xošev še ha-orex
on these articles$_i$ Dani thinks that the editor

ha-xadaš$_2$ efšar lešaxnea oto$_2$ levater ___$_1$
the new$_2$ it-is-possible to-convince him$_2$ to-give-up ___$_1$

(62) a. ha-orex ha-xadaš$_1$, dani xošev še al ha-ma'amarim
the editor the-new$_1$ Dani thinks that on the articles

haele$_2$ efšar lešaxnea oto$_1$ levater ___$_2$
these$_2$ it-is-possible to-convince him to-give-up

b. *et ha-orex ha-xadaš$_1$, dani xošev še ha-ma'amarim
ACC the-editor the-new$_1$ Dani thinks that the articles

haele$_2$ efšar lešaxnea ___$_1$ levater alehem$_2$
these$_2$ it-is-possible to-convince ___$_1$ to-give-up on-them$_2$

Using Fodor's (1978) terminology of fillers (F) and gaps (G) to refer to "preposed" constituents and "extraction" sites, the distribution of Fs, Gs and Ps (pronouns) in (61) and (62) are summarized in (63) and (64) respectively:

(63) a. F_1 F_2 G_2 P
b. *F_1 F_2 P G_1

(64) a. F_1 F_2 P G_2
b. *F_1 F_2 G_1 P

Notice that this pattern is exactly what our system here predicts: F_2 cannot bind P as long as there is an unbound gap G_1, i.e. as long as there is still an unretrieved quantifier-store. Therefore a sentence which has a distribution of gaps and pronouns as in (63b) or (64b) will only get an interpretation where P is a free pronoun. This would leave us with one filler too many, which explains the unacceptability of such a sentence. Notice that the explanation does not rely on left-right precedence and indeed any order of G_1 and P results in a starred configuration. (63a) and (64a) are acceptable configurations, since G_2 gets bound by F_2 before P has to be bound by F_1. By the time P has to get bound, the quantifier-store is empty and F_1 can bind P. Again, in this case, any order of G_2 and P is acceptable.

Engdahl (1979) has the following examples from Swedish:

(65) a. Haar ar flickorna$_1$ som jag inte minns vilka pojkar$_2$
lararen bad dem$_1$ dansa med ___$_2$

b. *Haar ar flickorna$_1$ som jag inte minns vilka pojkar$_2$
lararen bad dem$_2$ dansa med ___$_1$
"Here are the girls that I don't remember which boys the teacher asked them to dance with" (Engdahl's 13)

(65a) and (65b) exemplify the distribution in (63a) and (64b) respectively. Neither Engdahl, nor Maling and Zaenen have examples for (64b) or (63a). The only things mentioned about these cases are that in Swedish, "the resumptive pronoun always precede the gap" (Maling & Zaenen 1982) and "if the bindings are nested, a pronoun may not occur" (Engdahl 1979, p. 80). I conclude from these remarks that in Swdish, not only (64b) is starred, but so is (63a). The fact that (63a) is starred in Swedish requires an additional stipulation in my account, which shouldn't be surprising since this is a language specific phenomenon, and so it does in Engdahl's account.

Engdahl (1979) proposes a general parsing principle to account for (63b) and (64a)–(64b). She restates Fodor's (1978) Nested Dependency Constraint (NDC) as a general parsing strategy: "Associate the most recent filler with the next gap" (Engdahl's (22)). When the parser encounters a structure F_1-F_2-P-G …, the parsing strategy results in automatically assigning it F_1-F_2-P-G_2, i.e. the parser does not have to make a decision about which filler to associate with the gap.

> "… the NDC reduces the momentary processing load by only allowing the parser to make one assignment. Notice that the NDC enables the parser to resolve a pending filler-gap assignment locally and immediately. The closest filler is always associated with the next encountered gap. Most likely, this "local decision principle" will be highly valued by a parser engaged in real time processing."
> (Engdahl 1979, p. 84)

First notice that nothing about the NDC accounts for the fact that (63a) is starred in Swedish, since it is true of (63a) that the closest filler is associated with the gap. Secondly, as Engdahl herself notices, it is so far not at all clear whether this system is able to get all the semantic bindings right with only local decisions:

> "When the parser reaches a pronoun in a structure F_1-F_2-P …, it … has the option either to assume that it is a freely referring pronoun or that it is a resumptive pronoun, controlled by a preceding filler. At this stage in the processing, either choice may cause considerable reanalysis when more of the sentence is available."
> (Engdahl 1979, p. 85)

Thirdly, the NDC doesn't always make the right predictions in cases of multiple gaps. Engdahl (1980) has the following examples from Norwegian:

(66) a. Det var Eva_i laereren spurte $hvilken_2$ gutt vi trodde $\_\_\__2$ var spint pa $\_\_\__1$

b. Det var Eva_i laereren spurte $hvilken_2$ gutt vi trodde $\_\_\__1$ var spint pa $\_\_\__2$
"It was Eva that the teacher asked which boy we thought was mad at"
(Engdahl's 83)

The pattern of fillers-gaps exemplified in (66a) and (66b) are shown in (67a) and (67b) respectively:

(67) a. $F_1\ F_2\ G_2\ G_1$
b. $F_1\ F_2\ G_1\ G_2$

According to the NDC, (66b) and (67b) should be starred, since it involves crossing dependencies, i.e. binding of the second gap rather than the first to the last filler.

Other acceptable crossing dependencies occur in Icelandic:

(68) þessum krakka$_1$ herua geturdu aldrei imyndad þer hvada
this boy here you-can never guess what
gjof$_2$ eg gaf __$_1$ __$_2$
gift$_2$ I gave __$_1$ __$_2$ (Maling and Zaenen's 13c)

There are probably additional factors that influence the interpretation of crossing dependencies. Engdahl reports that nested readings, e.g. (66a), are strongly preferred in most contexts. But dependence upon context could hardly be accounted for by a principle about internal parsing of sentences.

Consider the following Hebrew sentences from Reinhart (1981a):

(69) a. al ha-nose ha-ze$_1$ ulay tuxal lomar li eize sfarim$_2$
on this topic$_1$ perhaps you-could tell me which books$_2$
ata xošev še keday li likro __$_2$ __$_1$
you think that it-is-worth to-me to-read __$_2$ __$_1$ (Reinhart's (14a))

b. hine sifri ha-rišon še oto$_1$ ani yodea ʿal eize
here is my book the-first that it$_i$ I know on which
nose$_2$ ata xošev še katavti __$_1$ __$_2$
topic$_2$ you think that I-wrote __$_1$ __$_2$ (Reinhart's (14b))

The sentences in (69a) and (69b) are both equally acceptable to me, but Reinhart reports that "this is the area where I found most disagreement in judment among the speakers I checked with" (p. 14). The disagreement though is about the status of what Reinhart calls "extraction across to S′ nodes", not about any difference in acceptability between (69a) and (69b). And indeed, examples where there is "extraction across one S′ node only" are cited as acceptable by Reinhart, even when they involve crossing dependencies:

(70) et ha-xavila ha-zot$_1$ hayiti roce la-daat im mi$_2$
ACC this package$_1$ I-would like to-know with whom$_2$
dan šalax __$_1$ le-rosa __$_2$
Dan sent __$_1$ to-Rosa __$_2$

Since word order in Hebrew VPs is sometimes relaxed (cf. Footnote 7), we should look at examples where the two gaps are not daughters of the same VP. Reinhart gives the examples in (71) and finds (71b), the one with crossing dependencies, unacceptable. For me, both are acceptable.

(71) a. et ha-sefer ha-ze$_1$, lo taamin et mi$_2$
ACC this book$_1$ you wouldn't believe ACC who$_2$
šixnati ___$_2$ lisxov ___$_1$ me-ha-sifriya
I-convinced ___$_2$ to-steal ___$_1$ from the-library

b. ?et ha-iš ha-ze$_1$, lo taamin eyze sefer$_2$
ACC this man$_1$ you wouldn't believe what book$_2$
šixnati ___$_1$ lisxov ___$_2$ me-ha-sifriya
I-convinced ___$_1$ to-steal ___$_2$ from-the-library
(Reinhart's 44a and 44b; 44b starred)

Where the two dislocated XPs do not share the same preposition or case marking, I do find the examples with crossing dependencies less acceptable:

(72) a. al ha-ma'amarim ha-ele$_1$, dani xoševše et ha-orex
on these articles$_1$ Dani thinks that ACC the-editor
ha-xadaš$_2$ 'efšar lešaxnea ___$_2$ le-vater ___$_1$
the-new$_2$ it-is-possible to-convince ___$_2$ to-give-up ___$_1$

b. ?et ha-orex ha-xadaš$_1$, dani xoševše al-ha-ma'amarim
ACC the editor the-new$_1$ Dani thinks that on the articles
ha-ele$_2$ efšar lešaxnea ___$_1$ le-vater ___$_2$
these$_2$ it-is-possible to-convince ___$_1$ to-give-up ___$_2$

The only thing we can conclude from this discussion of crossing dependencies is that their acceptability depends upon the language, the context, the speaker, and other structural properties of the sentences themselves. In any case, they seem to be a different phenomenon from the distribution of resumptive pronouns, for which this paper accounts.

We have seen problems that Engdahl's processing account for the distribution of resumptive pronouns runs into. Maling and Zaenen advocate a similar processing account, though they do not emphasize te NDC as an absolute principle. Rather, they suggest that whereas a gap increases "processing load" (cf. Wanner & Maratsos 1978), a resumptive pronoun does not. In other words, gaps interrupt the processing of a clause, since they have to be immediately paired with an antecedent on hold, whereas pronouns (resumptive or others) are not. In this respect, a resumptive pronoun is "preferable", specially in constructions involving crossing dependencies. This account, as it stands, does not make specific predictions as to what distributions of gaps and resumptive pronouns are acceptable. It also leaves open, just as Engdahl's did, the question of how resumptive pronouns are assigned to their antecedent.

Maling and Zaenen also propose an alternative syntactic solution. Their framework is basically that of Gazdar's plus allowing for multiply slashed categories. They propose the following metarule:

(73) A/B/C -→ A/C/$\underline{B}$ (Maling and Zaenen's 80)

where X/$\underline{X}$ is a resumptive pronoun.

We have already seen one problem in Maling and Zaenen's syntactic account, when we saw that it excluded crossing dependencies with gaps only (cf. 13).

We will now test each of the three falsifiable accounts at hand: Engdahl's processing account, Maling and Zaenen's syntactic account, and my semantic account, for their predictions to the case of sentences with three fillers. We first look for a case where each account makes a different prediction. Consider the following distribution:

(74) $F_1\ F_2\ F_3\ X_2\ X_3\ X_1$

What are the permissible values for X in (74)? The processing account predicts that the only permissible distribution of gaps and pronouns in (74) is:

(75) $F_1\ F_2\ F_3\ P_2\ G_3\ P_1$

since if we allowed G_2, it would get bound to F_2, and if we allowed G_1 to follow P_2–G_3, it would get bound to F_2.

The syntactic account predicts that the only permissible distribution of gaps and pronouns is the following:

(76) $F_1\ F_2\ F_3\ P_2\ G_3\ G_1$

since, under this account, we get a resumptive pronoun if and only if it replaces the first gap in a crossing dependency.

The semantic acount that I have presented in this paper allows for the following distribution:

(77) $F_1\ F_2\ F_3\ G_2\ G_3\ P_1$

since, by the time P_1 has to be bound by F_1, the quantifier-store will be empty, G_2 and G_3 having already been bound. Notice that (77) is not the only distribution I predict; (75) would be acceptable as well. But in order to show the superiority of my account, it is enough to find an example that exhibits the distribution in (77), and here it is:

(78) ze ha-iš$_1$ še od lo xatamta al-ha-mixtavim$_2$ še
this-is the man$_1$ that not yet you-signed on-the-letters that
etmol hexlatnu le-mi$_3$ anaxnu omdim
yesterday we-decided to-whom we are going
lišloax ___$_2$ ___$_3$ ito$_1$
to send ___$_2$ ___$_3$ with-him$_1$

I have not been able to check whether this example is grammatical in Norwegian.

5. Conclusion

This paper has shown that a treatment of the syntax and semantics of resumptive pronouns as distinct from the syntax and semantics of gaps has many advantages over non-distinct treatments. Syntactically, resumptive pronouns do not behave as gaps where the Coordinate Structure Constraint is concerned. Semantically, the antecedent of a resumptive pronoun has widest scope in the clause that contains the resumptive pronoun, whereas the antecedent of a gap does not. Another thing that the treatment in this paper accounts for is the distribution of resumptive pronouns in cases of multiple extractions.

References

Bach, Emmon & Robin Cooper. 1978. The NP-S analysis of relative clauses and compositional semantics. *Linguistics and Philosophy* 2. 1.

Bach, Emmon & Barbara Partee. 1980. Anaphora and semantic structure. Unpublished manuscript. University of Massachusetts, Amherst.

Borer, Hagit. 1979. Restrictive relative clauses in modern Hebrew. Unpublished manuscript, MIT.

Cooper, Robin. 1975. Montague's semantic theory and transformational syntax. Ph. D. dissertation. University of Massachusetts, Amherst.

Engdahl, Elisabet. 1979. The Nested Dependency Constraint as a parsing principle. In *Papers presented to Emmon Bach by his students*, ed. by E. Engdahl & M.J. Stein. University of Massachusetts, Amherst.

Engdahl, Elisabet. 1980. *The Syntax and Semantics of Questions in Swedish*. Ph.D. dissertation. University of Massachusetts, Amherst.

Fodor, Janet Dean. 1978. Parsing strategies and constraints on transformations. *Linguistic Inquiry* 9: 427–473.

Gazdar, Gerald. 1981. Unbounded dependencies and coordinate structure. *Linguistic Inquiry* 12: 155–184.

Gazdar, Gerald. 1982. Phrase structure grammar. *On the Nature of Syntactic Representations*, ed. G.K. Pullum & P. Jacobson. Dordrecht: Reidel.

Joshi, A.K. & L.S. Levy. 1977. Constraints on structural descriptions. *SIAM Journal of Computing.*

Joshi, A.K. & L.S. Levy. 1980. Phase structure trees bear more fruit than you would have thought. Paper presented at the *Annual Meeting of the Association for Computational Linguistics.*

Karttunen, Lauri & Stanley Peters. 1979. Conventional implicature. *Syntax and Semantics: Presupposition*, ed. by C.K. Oh & D. Dinneen, 1–56. San Diego: Academic Press.

Keenan, Edward. 1974. The Functional Principle: Generalizing the notion of "Subject of". *Proceedings of the 10th Meeting of the Chicago Linguistic Society*: 298–309.

Maling, Joan & Annie Zaenen. 1982. A base-generated account of Scandinavian extraction phenomena. *On the Nature of Syntactic Representations*, ed. G.K. Pullum & P. Jacobson. Dordrecht: Reidel.

Peters, Stanley. 1980. Paper presented at the Winter Meeting of the Linguistic Society of America.

Peters, Stanley. 1989. Handout to class lectures. University of Texas, Austin.

Reinhart, Tanya. 1978. Syntactic domains for semantic rules. *Formal Semantics and Pragmatics of Natural Languages*, ed. by F. Guenthner and S.J. Schmidt. Dordrecht: Reidel.

Reinhart, Tanya. 1981a. A second COMP position. *Theory of Markedness in Generative Grammar. Proceedings of the 1979 GLOW Conference*. 517–557. Pisa: Scuola Normale Superiore di Pisa.
Reinhart, Tanya. 1981b. Coreference and bound anaphora: a restatement of the anaphora question. Unpublished manuscript; Tel-Aviv University.
Wanner, E. & M. Maratsos. 1978. An ATN approach to comprehension. *Linguistic Theory and Psychological Reality*, ed. by Morris Halle, Joan Bresnan & George Miller. Cambridge, Mass.: MIT Press.

Appendix A

S1 a. $[_S$ NP VP[+pres]]
b. $[_S$ (NP) VP[-pres]]

S2 $[_{VP}$ V (XP_1 … XP_n)]
where XP_1 is NP or PP and XP_i = PP for $1< i \leq n$

S3 a. $[_{VP}$ V S′]
b. $[_{VP}$ V VP[-tns]]

S4 $[_{S'}$ COMP S]

S5 $[_{PP}$ P NP]

S6 $[_{NP}$ NP S′]

S7 $[_{NP}$ NP XP S]
where XP is NP or PP

S8 $[_S$ XP S]

Definition of "linked tree" (informal definition)

A "linked tree" is a tree with zero or more edges of a new kind (called "links") added to it, so that every node which is a link child, i.e. is at the bottom of a link, c-commands (with respect to the tree structure) its link parent(s). (from Peters (1981))[13]

---

13. Peters actually has an additional condition: "Every link child dominates (with rerspect to the tree structure) the link child(ren) of any link parent it dominates." The motivation for this condition is not clear to me, and, at least for Hebrew, it seems to be wrong, since the following is grammatical:

(vi) $[_S [_{NP}$ et hatisroket hazot] $[_S [_{VP} [_{Adv}$ od lo] $[_{VP} [_V$ macati] $[_{NP} [_{NP} [_{NP}$ sapar]
ACC hair-cut this yet not I-found hairdresser

$[_{S'} [_C$ še] $[_S [_{VP} [_V$ yodea] $[_{VP} [_V$ laasot …]]]]]] $[_{S'} [_C$ še] $[_S [_{VP} [_V$ yikba]
that knows to-do that fix

$[_{PP}$ li] $[_{NP}$ tor] $[_{Adv}$ hašavua]]]]]]]]]
to-me turn this-week

"I have not yet found a hair-dresser who can do this hair-cut who will give me an appointment this week."

Specification of "dislocated" constituents for Hebrew

A. An XP left sister of S or S′ is a "dislocated" constituent. A dislocated NP may be, and a dislocated PP necessarily is, a link child whose parents are dominated (with respect to the tree structure) by the S or S′ node to its right.

B. Only dislocated constituents may be link children.

Appendix B

Translation convention

Let X be a syntactic category.

A. A translation of X is a triplet $X' = \langle hX', qsX', rpsX' \rangle$ where:

(i) hX' ("head" of X') is the familiar Montague translation of X.

(ii) qsX' ("quantifier store" of X') is a set of pairs $\langle \alpha, i \rangle$, where α is of type $\langle\langle s, \langle e, t\rangle\rangle, t\rangle$ (i.e. the type of familiar translations of NPs) and i is a natural number.

(iii) $rpsX'$ ("resumptive pronoun store" of X') is a set of natural numbers.

B. $\langle hX', 0, 0 \rangle$ is a translation of X. Moreover, the only meanings of S are mappings of $\langle hS', 0, 0 \rangle$.

NP storage convention

A. If NP' is a translation of NP, then so is
$\langle {}^{\vee}p_i, qsNP' \cup \{\langle hNP', i\rangle\}, rpsNP' \rangle$.

B. Moreover, if $\langle \lambda PP\{x_i\}, 0, 0 \rangle$ is a translation of NP, then so are
$\langle \lambda PP\{x_i\}, 0, \{i\} \rangle$ and $\langle {}^{\vee}p_i, \langle \lambda PP\{x_i\}, i\rangle, \{i\} \rangle$.

Store retrieval convention

Let $\langle hS', qsS', rpsS' \rangle$ be a translation of S.

If $\langle \alpha_i, i \rangle \in qsS'$, then $\langle \alpha\, ({}^{\wedge}\lambda x_i\, [\lambda p_i hS']({}^{\wedge}\lambda PP\{x_i\})), qsS' - \{\langle \alpha, i\rangle\}, rpsS' \rangle$ is also a translation of S.

Translation rules

T1 a. $\langle hNP'\, ({}^{\wedge}hVP'), qsNP' \cup qsVP', rpsNP' \cup rpsVP' \rangle$

b. $\langle \lambda PP\{xi\}\, ({}^{\wedge}hVP'), qsVP', rpsVP' \rangle$

T2 $\langle hV'\, ({}^{\wedge}hXP'_1, \ldots, {}^{\wedge}hXPn'), \cup\, qsXPi', \cup\, rpsXPi' \rangle$

T3 a. $\langle hV'\, ({}^{\wedge}h\bar{S}', qs\bar{S}', rps\bar{S}' \rangle$

b. $\langle hV'\, ({}^{\wedge}hVP')), qsVP', rpsVP' \rangle$

T4 S'

T5 NP'

Let XP denote the immediate sister of S or S′ in S6, S7, and S8, and NP – the leftmost constituent in S7.

A. If $hXP' = {}^{\vee}p_i$ and $<\alpha, i> \in qsXP' \cap qsS'$, then

T6 $<\lambda P[\lambda p_i\, hS'](\hat{}\lambda R\alpha\, (P)), qsS' - \{<\alpha, i>\}, rpsS'>$
T8 $< [\lambda p_i\, hS']\, (\hat{}\, (\alpha)\, qsS' - \{<\alpha, i>\}, rpsS'>$

and if, moreover, $\alpha = PP\{x_i\}$, then

T7 $<[\lambda R\ NP']\, (\hat{}\lambda x_i\, [\lambda p_i hS']\, (\hat{}\alpha), qsNP' \cup qsS' -\{\{<\alpha, i>\}, rpsNP' \cup rpsS' - \{i\}>$

B. If $hXP' \neq {}^{\vee}p_{i,}$ $qsS' = 0$ and $i \in psS'$, then

T6 $<[\lambda RhXP']\, (\hat{}\lambda x_i\, hS'), qsXP' \cup rpsS' - \{i\}>$
T8 $<hXP'\, (\hat{}\lambda x_i\, hS'), qsXP', rpsXP' \cup rpsS' - \{i\}>$

C. Otherwise, T6, T7 and T8 are not defined.[14]

14. R in T6 and T7 is the variable introduced in Bach and Cooper (1978).