Resumption as Resource Management
(Dissertation Proposal)

Ash Asudeh

January 14, 2002

Contents

1 Introduction

This dissertation is about the syntax and especially the semantics of resumption, which can be characterized pretheoretically as follows:

(1) **Resumption (informal definition)**
The use of a referentially dependent element to mark a position which is syntactically and semantically associated with an element that occurs elsewhere in a sentence.

Even though the "referentially dependent element" in question is typically a pronoun, I use the term *resumption* rather than *resumptive pronoun* for a number of reasons. First, not all instances of resumption involve pronouns. Epithets, such as *the little bastard*, have been shown to function as resumptives in certain languages (Aoun et al. 2001).[1] Second, the term *resumptive pronoun* has in general been associated with pronouns functioning as the foot of a long distance dependency, i.e., $\overline{\text{A}}$ -movement in transformational terms (McCloskey 1979, 1990, to appear, Sells 1984). One of the major goals of the dissertation is to show that this definition of resumptive pronoun is too narrow and misses key generalizations.

For example, *copy raising* (Rogers 1971, 1972, 1974a,b, McCloskey and Sells 1988, Potsdam and Runner 2002) has not been analyzed as involving resumption. But notice that the use of the pronoun in (2) is resumptive under definition (1): sentence (2) is equivalent in meaning to (3).

(2) Richard seems like he reads a lot.

(3) It seems that/like Richard reads a lot.

The pronoun in (2) is marking the spot with which the name *Richard* is normally associated syntactically and the place in which it is interpreted.

In related examples, we see the same sort of alternation without a pronoun:

(4) Richard seems to read a lot.

(5) It seems that/like Richard reads a lot.

Examples (2)–(5) show that the same long distance relationship can be established between positions in a sentence with or without resumption.

Languages vary widely with respect to their use of resumption. Although English has a resumption strategy for copy raising, it is ungrammatical to use resumption for the target of control:

(6) Lisa tried to arrive early.

(7) *Lisa tried to she arrive early.

This follows from two facts about English. First, nonfinite verbs do not license overt subjects. Second, control in English involves syntactic identity of the controller and control target (Bresnan 1982a, Sag and Pollard 1991).

This contrasts with Serbo-Croatian[2] (Zec 1987), Farsi (Hashemipour 1985, 1989), and various other languages, where it is grammatical to use resumption in the corresponding environment. This is shown in the Serbo-Croatian sentence (8) and the Farsi sentences (9a) and (9b). Notice that both these languages exhibit pro-drop. In Serbo-Croatian, there is an obviation relation that holds between the controller and the pronominal control target, unless the pronoun is focused (Zec 1987). This means that the equivalent of (8) with an overt, unfocused pronoun is ungrammatical. Farsi differs in this respect, as it allows the pronominal control target to be realized overtly, although speakers prefer to drop it.[3]

[1] In what follows I will often lapse into talking about pronouns, but the reader should be aware that I mean to include epithets.
[2] This is the term that is used by Zec, my main literature source for this language.
[3] Actually, things are slightly more complicated: subject control verbs resist an overt control target more strongly than object control verbs.

(8) Petar je pokušao da dodje
Petar Aux tried Comp come(Pres)
Peter tried to come.
(Zec 1987: 142)

(9) a. Lisa sai kard ke zud biad.
Lisa try do.PAST.3SG COMP early arrive.SUBJUNCTIVE.3SG
Lisa tried to arrive early.

b. ?Lisa sai kard ke u zud biad.
Lisa try do.PAST.3SG COMP 3SG early arrive.SUBJUNCTIVE.3SG
Lisa tried to arrive early.

Serbo-Croatian and Farsi differ from English in having finite/subjunctive control complements, but there is a necessary *anaphoric control* relationship between the controller and the control target, as shown by the ungrammaticality of the following examples:

(10) a. *Lisa sai kard ke zud biam.
Lisa try do.PAST.3SG COMP early arrive.SUBJUNCTIVE.1SG

b. *Lisa sai kard ke man zud biam.
Lisa try do.PAST.3SG COMP 1SG early arrive.SUBJUNCTIVE.1SG

English also bars resumption in questions and relative clauses, which are environments in which resumptive pronouns are quite common cross-linguistically.

(11) **Questions**

a. *Which book does Richard like it?

b. Which book does Richard like?

(12) **Relative clauses**

a. *the book that Richard likes it

b. the book that Richard likes

Notice that the ungrammaticality of (11a) and (12a) is different from the ungrammaticality of (7), because there is no general injunction against a pronominal object of a transitive verb.

Irish, on the other hand, does allow resumption in these environments. The relative clause in (13) ends with the resumptive pronoun *í* ('her'); this pronoun is absent in (14). The form of the preverbal particle (glossed as *aN* or *aL*) is sensitive to the presence of resumption.

(13) **Resumption —** ***aN***
an ghirseach ar ghoid na síogaí í
the girl aN stole the fairies her
the girl that the fairies stole away
(McCloskey to appear: 7, (9b))

(14) **No resumption —** ***aL***
an ghirseach a ghoid na síogaí
the girl aL stole the fairies
the girl that the fairies stole away
(McCloskey to appear: 7, (9a))

Although resumption has been fairly widely studied in the literature, it is striking that no uniform treatment or explanation of the phenomenon has been offered. The formal theories in which resumption has been previously studied cannot generalize across the relevant cases and see no theoretical connection between them. In particular, copy raising, anaphoric control, and resumptive pronouns have received completely separate treatments. I use a recently developed formal theory of the syntax-semantics interface, *glue semantics* (Dalrymple 1999, 2001), to give a fully general account of resumption. Glue semantics allows this generalization because it is *resource-sensitive* and requires *resource management*. The syntactic framework I assume is Lexical Functional Grammar (LFG; Kaplan and Bresnan 1982, Bresnan 2001, Dalrymple 2001). I now turn to a brief exposition of resource-sensitivity. Glue semantics and resource management will be discussed further in section 2.

1.1 Resource-sensitivity at the syntax-semantics interface

Glue semantics is a theory of the syntax-semantics interface that uses linear logic for meaning composition. Linear logic is resource-sensitive, and this property is carried over to glue. What this means is that every meaningful element must be interpreted (i.e., composed with its functor or argument) exactly once. For example, sentence (15a) has one less occurrence of the adverb *allegedly* than (15b). However, the two sentences are truth-conditionally distinct, since the latter entails that the man is innocent, whereas the former does not. We cannot use the single occurrence of the adverbial *allegedly* twice to give (15a) the same meaning as (15b). Similarly, we cannot reuse the single negation in (16a) to derive the double negation semantics of (16b).

(15) a. This innocent man is allegedly guilty, according to some.

b. This allegedly innocent man is allegedly guilty, according to some.

(16) a. I don't like guava.

b. I don't not like guava.

Resource-sensitivity can be seen as a direct consequence of Frege's principle of compositionality: the meaning of the whole is the meaning of the parts and their mode of composition. However, according to the principle as commonly stated, and according to most semantic theories, there is no explicit injunction against composing the meanings of the parts multiple times. Glue semantics (like categorial grammar, with which it shares certain affinities) takes the property of resource-sensitivity to be a core property of language and makes it an explicit consequence of the logic.

The logic that glue uses for composition is *linear logic* (Girard 1987), an influential development in theoretical computer science. Linear logic is resource sensitive by definition. A glue meaning constructor pairs an expression from a meaning language ($\mathcal{M}$) with an expression in linear logic (G):

(17) $\mathcal{M} : G$

Meaning constructors are premises in linear logic proofs. Due to linear logic's resource accounting, premises cannot be arbitrarily disregarded or reused, like in classical logic.

This central property of linear logic and hence glue semantics will be the key to resumption. Resource-sensitivity means that all semantic resources must be accounted for in semantic composition. If there is some semantically contentful element that is not contributing its normal semantics, there must be another resource present that *manages* the unwanted resource. We will refer to these latter resources as *manager resources* (see section 2.3). We will see that this leads to a unified understanding of the occurrence of anaphors in resumption, including anaphoric control, copy raising, and resumptive pronouns.

I will also show that resource-sensitivity is the implicit basis for certain syntactic principles and sub-theories in the literature, such as Full Interpretation, No Vacuous Quantification, Theta Theory, the Bijection Principle, etc. With a resource-sensitive semantics, many of these can be weakened or dispensed with.

1.2 Goals of the dissertation

This dissertation attempts to answer the following central questions about the syntax and semantics of resumption:

1. What is resumption syntactically?
2. What is resumption semantically?
3. Why do languages use resumption?
4. Why are only referentially dependent elements (e.g., pronouns) used for resumption?
5. Why do some languages allow resumption in a given environment while others do not?

The answer to the first question is simple: syntactically, resumption is just the normal relationship between a pronoun and its antecedent.

The answer to the second question is apparently also simple, but is actually more complicated. The simple answer is that semantically resumption is nothing: the meaning of the pronoun is disregarded. For example, the Irish relative clause in (13), which contains a resumptive pronoun, is interpreted exactly like the relative clause in (14), which has no pronoun. The latter relative clause denotes a property, as the verb is missing an argument. If we disregard the pronoun, we can also assign the relative clause in (13) this interpretation, rather than the interpretation where the verb has picked up the pronoun and the clause is saturated, thus denoting a proposition. The complication is that whatever subsequently fills the argument position occupied by the pronoun must be the antecedent of the pronoun. Semantically, resumption is the use of a pronoun to establish an anaphoric dependency on some antecedent while leaving the argument that the pronoun corresponds to unresolved.

We can now answer question three. Languages use resumptions to 1) fulfill syntactic requirements while allowing certain interpretations, or 2) as one possible strategy to establish an anaphoric dependency between a dislocated element and a pronominal occupying the element's position as an argument to some predicate. The first case is demonstrated by anaphoric control, as in the Serbo-Croatian and Farsi examples discussed above, and copy raising, as in the English example (2) we began with. Sentence (3) shows that the pronoun in (2) is not necessary for purposes of interpretation. However, it is syntactically necessary, as its omission leads to ungrammaticality:

(18) *Richard seems like reads a lot.

The Irish relative clauses demonstrate the second case: the pronoun is not syntactically required, but resumption is one of two strategies available for indicating that a position is associated with some dislocated element.

Question four is probably the central question about resumption. This dissertation attempts to answer it at various levels. First, there is a formal answer, in terms of the linear logic for anaphors and manager resources. We will see in section 2.3 that an anaphor has the linear logic $A \multimap (A \otimes B)$, while a manager resource has $(A \otimes B) \multimap A$.[4] Together these yield a modifier on A, $A \multimap A$. The resource for the anaphor is removed by the manager resource. More importantly, *only* anaphors have are associated with a linear logic expression of the right form. Names, quantifiers, etc., have the wrong formal properties, in terms of linear logic, for manager resources to interact with them. The first answer to question four is that *anaphors are used for resumption due to their formal properties*.

But *why* do anaphors have this formal property? Precisely because they are anaphors. An anaphor must pick up its antecedent, but not consume it. Therefore, although an anaphor consumes its antecedent's resource to compute reference, it gives it back. As names are inherently referential and quantifiers are not referential, they do not have this property and have the wrong formal make up for manager resources. Thus, the formal answer is motivated in terms of the meaning of anaphors. Another way to look at this is that the linear logic side of glue meaning constructors is responsible for semantic composition; what the linear logic looks like is directly related to the meaning being composed. As a result, the second answer to question four is that *anaphors are used for resumption due to how they receive their meaning*.

[4] The symbol '$\multimap$' is linear implication, and the symbol '$\otimes$' is linear conjunction.

The third and final answer is related to the second one. The manager resource is essentially disposing of the anaphor, but allowing it to pick up its reference. This means that when the anaphor is disposed of, we are disposing of a meaning that we are guaranteed to have elsewhere, as there will be an antecedent. Thus, when an anaphor is disposed of, we are not losing any meanings, because its meaning is shared by some other element. If we were to dispose of a name or quantifier in a similar manner, we would be losing a substantive piece of the sentence's semantics. The third answer ist that *anaphors are used for resumption because they lack inherent meaning*.[5]

The answer to the fifth question is that languages differ in their ability to use resumption because there has to be something in the environment of the pronoun that removes the pronoun's meaning from consideration for interpretation (i.e., the manager resource). In the English case (2), it is the copy raising verb *seems*. In the Serbo-Croatian and Farsi cases (8) and (9), it is the control verb. In the Irish case (13), it is the preverbal particle *aN*. Thus, differences between languages with respect to resumption are reduced to differences in their lexical inventories. The hypothesis is that there must be an overt morpheme or construction associated with resumption. This contrasts sharply with McCloskey's (to appear) recent proposal that resumption signals the *mechanism* (MOVE or MERGE) that constructs the relevant CP.

1.3 Benefits of the resource management approach

We have already seen two of the principal benefits of the resource management approach to resumption. First, it yields a unified account of resumption that brings copy raising and anaphoric control into the fold, along with the usual case of resumptive pronouns. Second, there is a ready account of why it is always anaphoric elements that are related to resumption.

Third, we will see below that the analysis treats pronouns exactly like usual. This captures the remarkable generalization that the pronouns used in resumption are always the pronouns that would normally occupy the given argument position. That is, *no language has a special paradigm of resumptive pronouns*, as noticed by McCloskey, Sells, and many others.

Fourth, resumption must occur in specific environments, since it is dependent on the presence of manager resources. Manager resources are associated with specific lexical items (or possibly constructions). The analysis is therefore *lexicalist*.

Fifth, as a result the analysis treats typological variation for resumption as a matter of lexical inventories. Theories as diverse as Lexical Functional Grammar, Head-driven Phrase Structure Grammar (HPSG; Pollard and Sag 1994), and the Minimalist Program (Chomsky 1995) have converged on the desirability of locating language variation in the lexicon.

Sixth, glue semantics is a highly modular theory of the syntax-semantics interface, in a sense to be made precise shortly. I now turn to an introduction of the formal system of glue semantics.

2 Glue Semantics

Glue semantics embodies a notion of 'interpretation as deduction' closely related to categorial grammar's 'parsing as deduction'. Syntactic analysis of a sentence yields a set of glue meaning constructors, which essentially state how bits of meaning attach to words and phrases. A meaning constructor is a pairing of an expression from some meaning language and an expression of a glue logic, in our case a fragment of the resource-sensitive linear logic (Girard 1987). Resource-sensitivity models an important facet of natural languages and as a result reduces the space of possible derivations in a linguistically significant manner. Semantic derivations are linear logic deductions, with the meaning constructors serving as premises. The well-known Curry-Howard isomorphism systematically relates the meaning language to the linear logic.

This systematic relation between two distinct logics, the meaning language and the linear logic, makes glue a highly modular theory of the syntax-semantics interface, as mentioned above. By modular I mean framework-independent.

[5]This point was originally made to me by Mary Dalrymple (p.c.).

The meaning language can be any logic for semantics with lambda terms; for example there are existing versions of glue semantics using Montague's IL, DRT, and UDRT (Dalrymple et al. 1999b, Crouch and van Genabith 1999, van Genabith and Crouch 1999). Similarly, a variety of syntactic frameworks have been coupled with glue semantics, including LFG (Dalrymple 1999, 2001), Lexicalized Tree Adjoining Grammar (Frank and van Genabith 2001), and HPSG (Asudeh and Crouch 2001). This means that the results reported here should be readily applicable in other frameworks and that the theory developed here is not dependent on the correctness of one particular syntactic or semantic theory.

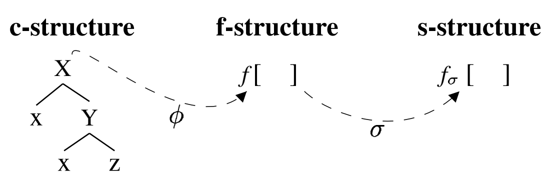
I will be using LFG as the syntactic framework and an appropriate version of higher-order intensional logic with events as the meaning language, although I will suppress event arguments and intensions in what follows.

This section provides an overview of glue semantics: its relationship to LFG syntax, the Curry-Howard isomorphism relating the meaning language side of the glue meaning constructor to the linear logic side, and how semantic composition works.

2.1 Glue and the Parallel Projection Architecture of LFG

Glue semantics is a general theory of the syntax-semantics interface and semantic composition. Although glue does not in any crucial way rely on an LFG grammatical architecture, it has been most developed within this architecture (Dalrymple 1999, 2001). LFG has a parallel projection architecture, which means that there are various levels of linguistic representation, called *projections*, present in *parallel*, and these projections are related by functional correspondences (also know as projection functions) which map elements of one projection onto elements of another (Kaplan 1987, 1995). This is a generalization of the original notion of functional correspondence in which the ϕ-function maps c(onstituent)-structures onto f(unctional)-structures (Kaplan and Bresnan 1982). As a result of this generalization, f-structures are mapped onto s(emantic)-structures by the σ-function (Dalrymple et al. 1999a). This results in an architecture like the following:

(19)



In terms of this architecture, glue semantics is a theory of the syntax-semantics interface (the σ-function) and semantic representation and interpretation (s-structure).

The various levels of grammatical representation are simultaneously present, but each level is governed by its own rules and representations. This separation of levels allows one to make simple theoretical statements about the aspect of grammar that the level in question models. Phrase structure, constituency, domination and linear order are represented at c-structure using trees, while grammatical functions, subcategorization, binding, raising, control, and various other aspects of syntax are represented at f-structure using attribute-value matrices. Semantics and the relationship between syntax and meaning are represented at s-structure using glue semantics: linear logic associated with a chosen meaning language.

Let us consider a very simple sentence in terms of LFG's projection architecture:

(20) Gonzo blushed.

Let us suppose we have only the simple annotated phrase structure rule in (21) and that *Gonzo* is of category NP and *blushed* is of category VP, with no further articulation of these phrasal categories. This would yield c-structure (22).

(21)
$$\begin{array}{llcc} \mathrm{S} & \longrightarrow & \mathrm{NP} & \mathrm{VP} \\ & & (\uparrow \textsc{subj}) = \downarrow & \uparrow = \downarrow \end{array}$$

(22)

$$\mathrm{S}_1$$

$(\uparrow \textsc{subj}) = \downarrow$	$\uparrow = \downarrow$
NP_2	VP_3
Gonzo	*blushed*

Given the usual interpretation of the variables '$\uparrow$' and '$\downarrow$' respectively as "the f-structure of the node above the node annotated $\uparrow$" and "the f-structure of the node annotated $\downarrow$", we can define f-structure (25) using the ϕ-function in (23). Note that we also use the ϕ-function to instantiate the feature descriptions $(\uparrow\ \textsc{subj}) = \downarrow$ and $\uparrow = \downarrow$, as in (24).

(23)
$$\phi(\mathrm{S}_1) = f$$
$$\phi(\mathrm{NP}_2) = g$$
$$\phi(\mathrm{VP}_3) = h$$

(24)
$$(\uparrow\ \textsc{subj}) = \downarrow$$
$$(\phi(\mathrm{S}_1)\ \textsc{subj}) = \phi(\mathrm{NP}_2)$$
$$(f\ \textsc{subj}) = g$$

$$\uparrow = \downarrow$$
$$\phi(\mathrm{S}_1) = \phi(\mathrm{VP}_3)$$
$$f = h$$

(25)
$$f,h\begin{bmatrix}\textsc{subj} & g\begin{bmatrix}\ \end{bmatrix}\end{bmatrix}$$

We use the lexical specifications of *Gonzo* and *blushed*, in (26), to further specify the c-structure as (27). This results in more information being added to the f-structure, as in (28).[6]

(26)

Gonzo	NP	$(\uparrow \textsc{pred}) =$ 'Gonzo'
		$gonzo : \uparrow_\sigma$
blushed	VP	$(\uparrow \textsc{pred}) =$ 'blush'
		$(\uparrow \textsc{tense}) =$ past
		$\lambda x.blush(x) : (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma$

(27)

$$\mathrm{S}_1$$

$(\uparrow \textsc{subj}) = \downarrow$	$\uparrow = \downarrow$
NP_2	VP_3
Gonzo	*blushed*
$(\uparrow \textsc{pred}) =$ 'Gonzo'	$(\uparrow \textsc{pred}) =$ 'blush'
$gonzo : \uparrow_\sigma$	$(\uparrow \textsc{tense}) = \textsc{past}$
	$\lambda x.blush(x) : (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma$

[6]Note that I have left off the label h. Since f-structure $f = h$, we can simply use the label f.

(28)
$$f\begin{bmatrix}\text{PRED} & \text{`blush'} \\ \text{SUBJ} & g\begin{bmatrix}\text{PRED} & \text{`Gonzo'}\end{bmatrix} \\ \text{TENSE} & \text{past}\end{bmatrix}$$

Finally, note that the lexical entries carry glue meaning constructors: a meaning language expression appears on the left of the uninterpreted symbol ':' and a linear logic expression appears on the right. The resources, or terms, in the linear logic expression are defined by the σ function. Thus, since $\uparrow$ on *Gonzo* refers to f-structure g, $\uparrow_\sigma$ is instantiated as g_σ. Similarly, since $(\uparrow \text{SUBJ})$ on *blushed* refers to g and $\uparrow$ refers to f (because $f = h$, by $\uparrow = \downarrow$), $(\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma$ is instantiated as $g_\sigma \multimap f_\sigma$. The fully instantiated meaning constructors are $gonzo : g_\sigma$ and $\lambda x.blush(x) : g_\sigma \multimap f_\sigma$. We consider glue semantics in more detail in the following section.

This simple example has hopefully clarified the idea of a projection architecture somewhat. An important feature of this architecture is that there can be systematic mismatches between grammatical levels. For example, null pronoun subjects in pro-drop languages are not present at c-structure, because they are unmotivated by the syntactic phenomena represented at that level. Rather, null pronouns are present at f-structure, where they can participate in agreement, binding, and other syntactic processes best represented at that level. Similarly, there can be systematic mismatches between f-structure and s-structure, and it is this aspect of the architecture that allows for an adequate semantics of control that nevertheless does not conflict with certain syntactic aspects of control, in particular argument sharing (Asudeh 2000, 2002).

2.2 Overview of Glue Semantics

Glue uses two logics: a meaning logic for representing meaning terms, and linear logic (Girard 1987) for assembling meanings (Dalrymple 1999). As already stated, linear logic is resource-sensitive: a linear logic proof is valid only if all premises are used exactly once. This is best exemplified by comparing classical propositional logic to propositional linear logic and observing the differences in certain entailment patterns. I use the standard symbols '$\rightarrow$' and '$\wedge$' for implication and conjunction in classical logic. The symbol '$\multimap$' is linear logic implication and '$\otimes$' is linear logic multiplicative conjunction.[7]

(29) Premise reuse

Classical Logic	Linear Logic
$A,\ A \rightarrow B \vdash B$	$A,\ A \multimap B \vdash B$
$A,\ A \rightarrow B \vdash B \wedge A$	$A,\ A \multimap B \nvdash B \otimes A$
Premise A reused, conjoined with conclusion B	Premise A is consumed to produce conclusion B, no longer available for conjunction with B

(30) Premise non-use

Classical Logic	Linear Logic
$A, B \vdash A$	$A, B \nvdash A$
Can ignore premise B	Cannot ignore premise B

Glue premises pair linear logic formulas (on the right of colons) with meaning terms (on the left). This is standard given the Curry-Howard Isomorphism (Howard 1980, Dalrymple et al. 1999b). In principle, we can choose any logic for the meaning logic, so long as a systematic relationship (such as the Curry-Howard) can be established between operations in the meaning language and those in the glue language (linear logic). Rules of inference determine how the meaning terms of premises are combined to give the meaning term of the conclusion. For example *modus ponens*

[7]There are various forms of conjunction that can be defined in linear logic, with distinct elimination and introduction rules. For reasons that would take us too far afield to consider, the multiplicative conjunction, or *tensor*, is standardly used in glue semantics (see Crouch and van Genabith 2000).

(implication elimination) corresponds to functional application of meaning terms, whereas hypothetical reasoning (implication introduction) leads to λ-abstraction, as shown:[8]

(31) **Implication Elimination** **Implication Introduction**

$$\dfrac{P : A \multimap B \qquad Q : A}{P(Q) : B}\multimap_{\mathcal{E}} \qquad\qquad \dfrac{\begin{array}{c}[Q : A]^i \\ \vdots \\ P : B\end{array}}{\lambda Q.P(Q) : A \multimap B}\multimap_{\mathcal{I},i}$$

Instances of implication elimination are indicated by the annotation '$\multimap_{\mathcal{E}}$' on the appropriate line of the proof. Implication introduction involves flagging an assumption in square brackets, and subsequently discharging this assumption once it has been used to prove another premise. In this case $[\mathrm{A}]^i$ is used to derive B, and we can discharge the assumption using implication introduction to get A $\multimap$ B, indicating the application of the proof rule and the discharged assumption by '$\multimap_{\mathcal{I},i}$'.

The following simple example shows the natural deduction rules and Curry-Howard isomorphism working together to prove that $A \multimap B \vdash A \multimap B$.

(32)

$$\dfrac{\dfrac{[Q : A]^i \qquad P : A \multimap B}{P(Q) : B}\, function\ application, \eta\text{-}equivalence : \multimap_{\mathcal{E}}}{\lambda Q.P(Q) : A \multimap B}\, lambda\ abstraction : \multimap_{\mathcal{I},i}$$

In the first step, $Q : A$ is assumed (indicated by square brackets) and the assumption is flagged with the superscript i. We take this assumption and combine it with our one premise $A \multimap B$ by elimination, which corresponds to function application in the meaning language. The assumption is discharged in the second step, reintroducing the assumed linear logic atom A. On the meaning language side this corresponds to abstracting over the associated variable, Q.

To simplify proofs somewhat, glue conjunction ($\otimes$) in the antecedent of an implication will be cashed out as implication, by the following equivalence (which also holds for classical propositional logic):

(33) $(A \otimes B) \multimap C \equiv B \multimap (A \multimap C)$

For example, we will see shortly that the manager resources that handle resumption are written as

(34) $(A \otimes B) \multimap A$

This is equivalent to

(35) $(B \otimes A) \multimap A$

which is in turn equivalent to

(36) $A \multimap (B \multimap A)$

After we have considered the interaction of pronouns and manager resources in section 2.3, manager resources will be written as in (36), starting in section 3.

As another example, the glue semantics for a transitive verb can be written using a conjunction, such that the verb consumes the resources of its subject and object to give its meaning, or it can be written such that the verb consumes its object's resource and then its subject's resource to give its meaning:

[8] I will present my glue proofs in the natural deduction (ND) style (Dalrymple 2001). For glue proofs in the sequent style familiar from categorial grammar, see Dalrymple et al. (1999a). For a more thorough overview of linear logic in linguistic theory, proof theory and proof rules in both natural deduction and sequent styles, see Crouch and van Genabith (2000).

(37) An example: transitive verbs
$((\uparrow \text{SUBJ})_\sigma \otimes (\uparrow \text{OBJ})_\sigma) \multimap \uparrow_\sigma \equiv (\uparrow \text{OBJ})_\sigma \multimap ((\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma)$

Using the equivalence in (33) with the equivalence in (38), the subject can be consumed first, as in (39).

(38) $(\text{A} \otimes \text{B}) \equiv (\text{B} \otimes \text{A})$

(39)
$$\begin{aligned} &(\uparrow \text{OBJ})_\sigma \multimap ((\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma) \\ \equiv\ &((\uparrow \text{SUBJ})_\sigma \otimes (\uparrow \text{OBJ})_\sigma) \multimap \uparrow_\sigma \\ \equiv\ &((\uparrow \text{OBJ})_\sigma \otimes (\uparrow \text{SUBJ})_\sigma) \multimap \uparrow_\sigma \\ \equiv\ &(\uparrow \text{SUBJ})_\sigma \multimap ((\uparrow \text{OBJ})_\sigma \multimap \uparrow_\sigma) \end{aligned}$$

However, we cannot simplify all conjunctions like this. Pronouns in particular, which are the subject of much of this work, have meaning constructors of the form $\lambda y.y \times y : A \multimap (A \otimes B)$. This representation for pronouns is essentially that of Dalrymple et al. (1999b), adapted to the Curry-Howard notation.[9] The linear logic side of the meaning constructor has a conjunction in the consequent of an implication, for which the equivalence (33) does not hold. The meaning language contains a pair formed by the product operator '$\times$', which essentially forms an ordered pair out of its operands. We therefore need proof rules for linear logic conjunction, as well:

(40) **Conjunction Introduction** **Conjunction Elimination**

$$\frac{P : A \qquad Q : B}{P \times Q : A \otimes B}\otimes_{\mathcal{I}} \qquad\qquad \frac{P : A \otimes B \qquad \begin{array}{c}[X : A]^i\ [Y : B]^j \\ \vdots \\ R : C\end{array}}{let\ P\ be\ X \times Y\ in\ R : C}\otimes_{\mathcal{E},i,j}$$

I will not use conjunction introduction, as it is generally easier to deal with the conjuncts separately, but I will use conjunction elimination for pronouns. Notice the use of the predicate *let* in the lambda calculus in the meaning language side of this proof rule. This is pairwise substitution: the meaning language correspondents of the conjuncts are substituted as an ordered pair for a pair of assumed variables in some predicate. The *let* can be reduced by standard β-reduction of lambda terms.

Let us do a proof for a simple sentence involving a pronoun to see an application of conjunction elimination.

(41) Dee hurt herself.

(42)

$$\dfrac{\dfrac{\dfrac{dee : A \qquad \lambda z.z \times z : A \multimap (A \otimes B)}{dee \times dee : A \otimes B}\multimap_{\mathcal{E}} \qquad \dfrac{[Y : B]^2 \qquad \dfrac{[X : A]^1 \qquad \lambda x \lambda y.hurt(x,\ y) : A \multimap (B \multimap C)}{\lambda y.hurt(X,\ y) : B \multimap C}\multimap_{\mathcal{E}}}{hurt(X,\ Y) : C}\multimap_{\mathcal{E}}}{let\ dee \times dee\ be\ X \times Y\ in\ hurt(X,\ Y) : C}\otimes_{\mathcal{E},1,2}}{hurt(dee, dee) : C}\beta\text{-}reduction$$

On the left side, the pronoun *herself* picks up its antecedent, yielding the conjunction $dee \times dee : A \otimes B$. On the right side, we make two assumptions, $[X : A]$ and $[Y : B]$, and combine these with the predicate $\lambda x \lambda y.hurt(x,\ y) : A \multimap (B \multimap C)$, eventually yielding $hurt(X,\ Y) : C$. This result is combined with the pronoun meaning via the conjunction elimination rule. Finally, we reduce the *let* by substituting dee for X and dee for Y, yielding $hurt(dee, dee) : C$, which is an appropriate meaning for (41).

[9] Notice that this representation does not handle cross-sentential anaphora, although the generalized system in Dalrymple (2001) does. However, the phenomenon of resumption is strictly intra-sentential, therefore we will not need the generalized system for anaphora, although these results can be straightforwardly translated into that system.

2.3 Resource management

Resource-sensitivity means that all semantic resources must be accounted for in semantic composition. If there is some semantically contentful element that is not contributing its normal semantics, there must be another resource present that *manages* the unwanted resource. Let us call these latter resources *manager resources*. This leads to a unified understanding of the occurrence of anaphors in resumption, including anaphoric control, copy raising, and resumptive pronouns.

As we just observed, anaphoric elements in this theory will be represented as follows in the linear logic, where A is the resource corresponding to the antecedent and B is the resource corresponding to the anaphor.

(43) $A \multimap (A \otimes B)$

The antecedent is consumed to produce the anaphor and another copy of the antecedent.

In the meaning language this will correspond to the anaphor picking up the reference of the antecedent. Pronouns do not add any information to their antecedent, having a representation like

(44) $\lambda Z.Z \times Z : A \multimap (A \otimes B)$

Suppose the antecedent of this pronoun were the proper name *Kim*, which contributes a meaning constructor like $\mathit{kim} : A$. Combining this with the pronoun would yield:

(45) $\mathit{kim} \times \mathit{kim} : A \otimes B$

When we eliminate the conjunction, we have two kim meanings, one for the antecedent and one for the pronoun.

Epithets are slightly more complicated. They contribute two meaning constructors. The first meaning constructor resolves the anaphoric dependency and looks identical to that of pronouns. The second meaning constructor is a modifier on the predicate to which the epithet is an argument. The denotation of the predicate is a property. The epithet takes this property and returns a conjunction of the property and the property of the speaker being either positively disposed ($\mathit{disposition}^{+}$) or negatively disposed ($\mathit{disposition}^{-}$) to the referent of the epithet. For example, the epithet *the little darling*, used unironically, would contribute meaning constructors like

(46) $\lambda Z.Z \times Z : \multimap A \multimap (A \otimes B)$

$\lambda P \lambda x.P(x) \wedge \mathit{disposition}^{+}(\mathit{speaker},\ x) : (B \multimap C) \multimap (B \multimap C)$

In order to see how this works, let us briefly consider the complement in the following sentence.

(47) Kim's mother knows the little darling won.

The first meaning constructor is resolved to $\mathit{kim} \times \mathit{kim} : A \otimes B$. The second meaning constructor consumes the meaning of the predicate *won*, $\lambda y.\mathit{win}(y)$, and returns the property $\lambda x.\mathit{win}(x) \wedge \mathit{disposition}^{+}(\mathit{speaker},\ x)$. In our example the reference of the epithet has been resolved, and eliminating the conjunction gives the meaning $\mathit{kim} : A$ for the antecedent and $\mathit{kim} : B$ for the epithet. Combining the latter with the conjoined property yields the proposition $\mathit{win}(\mathit{kim}) \wedge \mathit{disposition}^{+}(\mathit{speaker},\ \mathit{kim})$, which is true if and only if Kim won and the speaker is positively disposed toward Kim.

It is the meaning constructor that epithets and pronouns share that is the key to their behaviour as resumptive elements. The manager resources that deals with the anaphoric resources have the following linear logic:

(48) $(A \otimes B) \multimap A$

It should be fairly obvious that the antecedent of the manager resource is the consequent of the anaphor resource. By transitivity, an anaphor and a manger resource are together equivalent to a modifier on A:

(49) $A \multimap (A \otimes B), (A \otimes B) \multimap A \vdash A \multimap A$

We can therefore formally define resumption as follows:

(50) **Resumption (formal definition)**
Resumption is the combined use of a manager resource, $(A \otimes B) \multimap A$, and an anaphoric resource, $A \multimap (A \otimes B)$. Resumption yields a modifier, $A \multimap A$, on the antecedent, A.

The details of how this work will become clearer shortly. The point is simply that these two types of resources together result in the removal of the anaphor's resource, but the anaphor is allowed to resolve its reference. This means we retain the anaphoric dependency that the pronoun encodes, while disposing of the actual pronominal resource.

3 Anaphoric Control

Various otherwise typologically diverse languages, such as Serbo-Croatian, Greek, Bulgarian, Albanian, Romanian (which are all Balkan languages), and Farsi, have control structures where the control verb takes a finite/subjunctive complement. Crucially, evidence suggests that the controlled complement has a syntactically active subject[10] in argument position.

If we wish to maintain a property semantics for the controlled complement, like Chierchia (1984a,b) has argued for, there must be something in the environment of the complement that prevents the controlled pronominal subject from composing with the rest of the complement, thus preventing construction of a propositional semantics and ensuring that the complement denotes a property. In other words, a manager resource is required to consume the pronoun, effectively changing the type of the complement from a proposition to a property.

Let us consider the following Serbo-Croatian sentence:

(51) Petar je pokušao da dodje
Petar Aux tried Comp come(Pres)
Peter tried to come.
(Zec 1987: 142)

At c-structure, the clausal complement (IP) does not have a subject, but there are reasons for supposing that the complement to the matrix verb has a subject at f-structure, despite not having an overt subject constituent at c-structure.

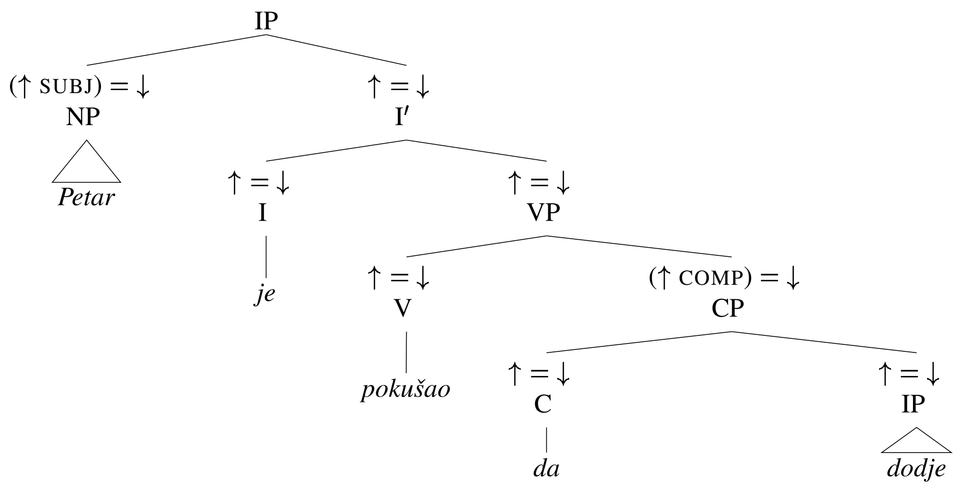
There is an overt complementizer, indicating that this is a CP, not a VP. Since it is a finite CP, it must have a subject for theoretical reasons. Zec also provides empirical evidence that there is actually a null pronominal subject in the embedded clause. So there is an f-structure SUBJ. Control in languages with finite control (i.e., where the complement clause has a subject at f-structure and the complement has a finite tense), is modeled as *anaphoric control* in LFG, whereby there is an anaphoric binding relationship between the controller and a controllee that is pronominal (i.e., it has PRED 'pro').

Pokušao is nevertheless a control verb and there is a relationship of obligatory control between its subject and the subject of its complement. Furthermore, Serbo-Croatian control verbs participate in certain inference patterns initially explored by Chierchia (1984a,b) that indicate that the clausal control complement denotes a property despite initial appearances (Zec 1987). This is a problem for the particular approach that Chierchia himself adopts, because his theory predicts that the sentential complement in (51) should denote a proposition, as it has a subject. I have shown elsewhere (Asudeh 2000, 2002) that LFG's parallel projection architecture and glue semantics can overcome this problem, and in this section I will review this solution.

I assume the following c-structure for sentence (51):

[10] The controlled subject may be either null or overt, depending on the language.

(52)



The lexical terminals for the c-structure are given in (53), with the f-structure nodes instantiated to the f-structure in (54) below.[11]

(53) *Petar* N $(\uparrow \textsc{pred}) = \text{`Petar'}$

$$\begin{aligned} petar :\ & \uparrow_\sigma \\ & = g_\sigma \end{aligned}$$

je I $(\uparrow \textsc{tense}) = \text{past}$

pokušao V $(\uparrow \textsc{pred}) = \text{`try'}$
$((\uparrow \textsc{comp subj})_\sigma\ \textsc{antecedent}) = (\uparrow \textsc{subj})_\sigma$

$$\begin{aligned} \lambda w \lambda P.try(w, P) :\ & (\uparrow \textsc{subj})_\sigma \multimap (((\uparrow \textsc{comp subj})_\sigma \multimap (\uparrow \textsc{comp})_\sigma) \multimap \uparrow_\sigma) \\ & = g_\sigma \multimap ((i_\sigma \multimap h_\sigma) \multimap f_\sigma) \end{aligned}$$

$$\begin{aligned} \lambda x \lambda y.x :\ & ((\uparrow \textsc{comp subj})_\sigma\ \textsc{antecedent}) \multimap \\ & \quad ((\uparrow \textsc{comp subj})_\sigma \multimap ((\uparrow \textsc{comp subj})_\sigma\ \textsc{antecedent})) \\ & = g_\sigma \multimap (i_\sigma \multimap g_\sigma) \end{aligned}$$

da C $(\uparrow \textsc{mood}) = \text{decl}$

dodje V $(\uparrow \textsc{pred}) = \text{`come'}$
$(\uparrow \textsc{tense}) = \text{pres}$

$$\left(\begin{aligned} & (\uparrow \textsc{subj pred}) = \text{`pro'} \\ & \lambda y.y \times y :\ (\uparrow_\sigma\ \textsc{antecedent}) \multimap (\uparrow_\sigma \otimes (\uparrow_\sigma\ \textsc{antecedent})) \\ & \qquad = g_\sigma \multimap (i_\sigma \otimes g_\sigma) \end{aligned} \right)$$

$$\begin{aligned} \lambda z.come(z) :\ & (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma \\ & = i_\sigma \multimap h_\sigma \end{aligned}$$

Notice that the verb *dodje* ('come') optionally specifies the PRED of its subject as 'pro'. This is the standard analysis of pro-drop in LFG. In other words, it is a general fact about Serbo-Croatian that the verb can provide its subject information, with no overt realization of this subject. As usual the verb is still looking to consume its subject's resource.

[11] Although I assume that the auxiliary verb *je* contributes tense information and this presumably has semantic import, I do not deal with this factor here.

Therefore, when it provides the subject's PRED, the verb must also provide its pronominal meaning constructor. The pronoun's meaning constructor has a linear conjunction ($\otimes$) on the right hand side (see footnote 12 below). In the meaning language, this corresponds to a pair (as indicated by the product operator, '$\times$') of the pronoun and its antecedent, which are identical. At first glance it seems that this optional specification of the pronoun renders the pronoun consumption premise in the control verb superfluous, as the complement of a control could just not realize the pronominal information. But this would miss a generalization, as Zec (1987) has argued that this is a case of pro-drop, with a syntactically active subject at f-structure.

Thus, we get the following f-structure for (51), with a pronominal subject in the controlled complement:

(54)
$$f\left[\begin{array}{ll} \mathrm{PRED} & \text{`try'} \\ \mathrm{SUBJ} & g\left[\begin{array}{ll} \mathrm{PRED} & \text{`Petar'} \end{array}\right] \\ \mathrm{COMP} & h\left[\begin{array}{ll} \mathrm{PRED} & \text{`come'} \\ \mathrm{SUBJ} & i\left[\begin{array}{ll} \mathrm{PRED} & \text{`pro'} \end{array}\right] \\ \mathrm{MOOD} & \text{decl} \end{array}\right] \\ \mathrm{TENSE} & \text{past} \end{array}\right]$$

The control verb *pokušao* ('try') specifies that its subject is the antecedent of its complement's subject, via the functional equation $((\uparrow \mathrm{COMP\ SUBJ})_\sigma\ \mathrm{ANTECEDENT}) = (\uparrow \mathrm{SUBJ})_\sigma$, thus establishing the anaphoric control relation.

This contrasts with the f-structure for the English equivalent of (51), *Petar tried to come*. The controller functionally controls and is identical to the control target:

(55)
$$f\left[\begin{array}{ll} \mathrm{PRED} & \text{`try'} \\ \mathrm{SUBJ} & g\left[\begin{array}{ll} \mathrm{PRED} & \text{`Petar'} \end{array}\right] \\ \mathrm{COMP} & h\left[\begin{array}{ll} \mathrm{PRED} & \text{`come'} \\ \mathrm{SUBJ} & \\ \mathrm{MOOD} & \text{decl} \end{array}\right] \\ \mathrm{TENSE} & \mathrm{PAST} \end{array}\right]$$

There is substantial evidence for supposing that English *try* is a *functional control* verb, rather than an anaphoric control verb (Bresnan 1982a, Falk 2001).

Beside the languages mentioned already, there are languages such as Icelandic which have nonfinite control complements (like English) but which nevertheless display evidence of anaphoric control. Icelandic in particular shows lack of case agreement between a controller and a control target that would be assigned quirky case (Andrews 1982). In other words, the control target would be assigned quirky case by its predicate, but the controller bears structural case. This would be impossible under functional control, as the values of the two grammatical functions are identical and cannot vary for the case feature.

Despite the syntactic differences at f-structure, the linear logic side of the glue meaning constructors for English and Serbo-Croatian control verbs are formally very similar.

(56) Comparison of the English and Serbo-Croatian control verbs

tried V ($\uparrow$ PRED) = 'try'

$$\lambda x \lambda P.try(x, P) : (\uparrow \text{SUBJ})_\sigma \multimap (((\uparrow \text{XCOMP SUBJ})_\sigma \multimap (\uparrow \text{XCOMP})_\sigma) \multimap \uparrow_\sigma)$$
$$= g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma)$$

pokušao V ($\uparrow$ PRED) = 'try'

$$\lambda x \lambda P.try(x, P) : (\uparrow \text{SUBJ})_\sigma \multimap (((\uparrow \text{COMP SUBJ})_\sigma \multimap (\uparrow \text{COMP})_\sigma) \multimap \uparrow_\sigma)$$
$$= g_\sigma \multimap ((i_\sigma \multimap h_\sigma) \multimap f_\sigma)$$

There are two key similarities. First, both verbs denote relations between individuals and properties. Second, each control verb consumes the entire dependency that corresponds to its controlled complement.

The key difference is that the semantics of functional control in English requires no manager resource, whereas anaphoric control in Serbo-Croatian does require this additional resource. In English the controlled subject is identical to the control verb's subject at f-structure, as indicated schematically by the solid line in (57). There is one only one resource contributed by this shared subject, and it is consumed by the control verb. Although the controlled complement also depends on this subject, the English control verb consumes the entire complement dependency, and all resources are managed satisfactorily.

(57) *try*
$g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma)$

In Serbo-Croatian the controller and controlled subject are not identified, as indicated by the dashed line in (58).

(58) *pokušao*
$g_\sigma \multimap ((i_\sigma \multimap h_\sigma) \multimap f_\sigma)$

As discussed above, Serbo-Croatian control involves a pro-dropped subject in the controlled complement. This subject is contributed by the complement verb, which also contributes its resource. In general the verb must contribute the subject's resource in order for the semantics of pro-drop to proceed as expected. Like the English control verb, the Serbo-Croatian control verb consumes its own subject and the dependency of the controlled complement's verb on the controlled subject. However, as the complement verb is the only consumer of the controlled subject, there is nothing left to consume the controlled subject's resource. The controlled subject requires resource management, and this is why the verb *pokušao* contributes a manager resource:

(59) Manager resource, from control verb *pokušao*
$\lambda x \lambda y.x : g_\sigma \multimap (i_\sigma \multimap g_\sigma)$

This meaning constructor is motivated because the pronoun can only be bound by the controlling subject. The pronominal resource that the pronoun contributes is consumed by the manager resource. Although the pronoun contributes a normal pronominal meaning, the presence of the pronoun and the resource manager together is, in effect, a device that is employed to establish a control equation in certain languages, such as Serbo-Croatian (Zec 1987) and other Balkan languages, Icelandic (Andrews 1982), and Farsi. Thus, we can think of this glue formula as going hand in hand with the anaphoric control equation, just as the embedded verb's contribution of its null subject goes hand in hand with a glue formula for that subject. The first glue formula, which gives the semantics of the control verb, is the same as the glue for the control verb *try* in English, a language that uses functional control in the syntax. In both cases, the matrix subject is consumed to yield an implication that consumes the *entire* clausal complement to yield the semantics of the outer f-structure and thus the sentence. The typological difference between obligatory anaphoric control and functional control in the syntax is reflected by a typological difference in the glue language, while allowing the same denotation in the meaning language, or semantics proper.

The following proof demonstrates the parallelism. Notice that once the glue premises dealing with the anaphoric control relation and embedded subject have been consumed, the remainder of the proof (starting at the fourth line) is identical to the proof in (61) for the English sentence *Gonzo tried to go*.[12]

(60)

$$\dfrac{\lambda z.come(z) : i_\sigma \multimap h_\sigma \qquad \dfrac{\lambda w \lambda P.try(w,P) : g_\sigma \multimap ((i_\sigma \multimap h_\sigma) \multimap f_\sigma) \qquad \dfrac{\dfrac{\dfrac{petar : g_\sigma \quad \lambda y.y \times y : g_\sigma \multimap (i_\sigma \otimes g_\sigma)}{petar : i_\sigma,\ \ petar : g_\sigma}{\multimap_{\mathcal{E}}, \otimes_{\mathcal{E}}} \qquad \lambda x \lambda y.x : g_\sigma \multimap (i_\sigma \multimap g_\sigma)}{petar : i_\sigma,\ \ \lambda y.x : i_\sigma \multimap g_\sigma}{\multimap_{\mathcal{E}}}}{petar : g_\sigma}{\multimap_{\mathcal{E}}}}{\lambda P.try(petar, P) : (i_\sigma \multimap h_\sigma) \multimap f_\sigma}{\multimap_{\mathcal{E}}}}{try(petar, \lambda z.come(z)) : f_\sigma}{\multimap_{\mathcal{E}}}$$

(61)

$$\dfrac{\lambda z.go(z) : g_\sigma \multimap h_\sigma \qquad \dfrac{\lambda w \lambda P.try(w,P) : g_\sigma \multimap ((g_\sigma \multimap h_\sigma) \multimap f_\sigma) \qquad gonzo : g_\sigma}{\lambda P.try(gonzo, P) : (i_\sigma \multimap h_\sigma) \multimap f_\sigma}{\multimap_{\mathcal{E}}}}{try(gonzo, \lambda z.go(z)) : f_\sigma}{\multimap_{\mathcal{E}}}$$

The intuition behind this analysis is that although there is a null pronominal subject in the clausal complement of an anaphoric control verb, it is only there to maintain the obligatory syntactic relationship between the controller and the controllee. Thus, there is typological variation in the syntactic mechanisms used to maintain control relationships. English uses functional control, while other languages, such as Serbo-Croation and Farsi, use obligatory anaphoric control. The differences in the syntactic mechanisms are reflected in the glue semantics, but it is still possible for the controlled clausal complements to have the same denotation, no matter which syntactic mechanism the grammar of a language employs. Therefore, there is a separation of syntax and semantics, but there is a systematic relationship between the two, and there can be typological differences in one, without there *necessarily* being differences in the other.

4 Copy Raising and Physical Perception Verbs

Potsdam and Runner (2002) have recently argued for a base-generation account of copy raising (CR), within the assumptions of the Minimalist Program (Chomsky 1995). They take CR to involve sentences like (62) and (63).

(62) Richard seems like he won.

(63) Richard looks like he won.

The assumption that verbs like *seems* and verbs like *looks* are undifferentiated may seem viable based on simple examples like these. However, if we examine more data, we see that there are differences:

(64) Some contrasts to consider:

[12] This proof makes implicit use of conjunction elimination ($\otimes_{\mathcal{E}}$). Rather than using the proof rule (given in (40) in section 2 above), the conjuncts are separated immediately and carried through the proof. This is a notional shorthand, and the reader can check that proper use of conjunction elimination gives the same result. The proof rule is used in full in sections 4.2 and 5.

Phenomenon	*seems*	*looks*
(a) Raising (infinitival comp.)	Richard seems to have won.	%Richard looks to have won.
(b) Raising (predicative AP)	Richard seems sad.	Richard looks sad.
(c) No raising	It seems that Richard won.	*It looks that Richard won.
(d) Copy raising	Richard seems like he won. Every boy seems like he won.	Richard looks like he won. Every boy looks like he won.
(e) No copy raising	It seems like Richard won.	It looks like Richard won.
(f) Copy Raising w/ non-subject pronoun	Richard$_i$ seem like you can rely on him$_{i/*j}$.	Richard$_i$ looks like you can rely on him$_{i/j}$.
(g) Copy Raising w/o pronoun	*Richard seems like Emma won. *Every boy$_i$ seems like she$_j$ won.	Richard looks like Emma won. Every boy$_i$ looks like she$_j$ won.

In particular, note that *seems* and *looks* differ in their ability to take an expletive under extraposition (65) and the necessity of having a pronominal bound to their subjects (66).

(65) a. It seems that Richard won.

b. *It looks that Richard won.

(66) a. *Richard seems like Emma won.

b. Richard looks like Emma won.

This data indicates that *seems* does not have a thematic subject and can thus appropriately be called a copy raising verb, because it shares this definitional property with subject-to-subject raising verbs. On the other hand, *looks* cannot take an expletive subject and does not even need a pronominal target in its complement; it is not clear in what sense *looks* in these sentences could be called a raising verb. Therefore, I will follow Rogers (1971, 1972, 1974a,b) in calling *looks* a *physical perception verb*. An account such as Potsdam and Runner (2002), which relies on an base-generated anaphoric dependency between the copy raising verb's subject and its complement's subject, cannot straightforwardly capture the physical perception verbs or the differences between copy raising verbs and physical perception verbs. Nonetheless, movement accounts of copy raising run into the same problems and more.

Potsdam and Runner (2002) argue that movement analyses, such as Ura's (1998), are problematic because copy raising would be movement from a Case position, thus violating Last Resort (Chomsky 1993), and would also violate the Tensed S Condition (Chomsky 1973), which prohibits movement from an argument position in a tensed clause. Potsdam and Runner further argue that their approach correctly predicts that a quantifier subject of a copy raising verb must take wide scope with respect to the CR verb (Lappin 1984), in contrast to raised quantifiers in subject-to-subject raising (SSR) or subject-to-object raising (SOR), which can also take narrow scope with respect to the raising verb. The available readings in copy raising are illustrated in (67) (Potsdam and Runner 2002: (35)).

(67) a. Two people seem like they have won the lottery. *seem $>$ 2, 2 $>$ seem

b. $\neq$ It seems like two people have won the lottery. seem $>$ 2

c. $=$ Two people are such that they seem like they have won the lottery. 2 $>$ seem

Potsdam and Runner write, "Under the base-generation analysis, the DP cannot be interpreted in the embedded clause . . . because Quantifier Lowering is blocked by the pronominal copy. The pronoun, not being a trace, prevents reconstruction." However, Aoun et al. (2001) have argued that resumptive pronouns and epithets in Lebanese Arabic do allow reconstruction, thus not being a trace is not a sufficient condition, under these theoretical assumptions, for disallowing reconstruction.

[12] This sentence is available in some dialects and registers (e.g., in sports commentary) with *looks*, but it becomes much worse if we substitute similar verbs, such as *smells*.

The Lebanese Arabic facts do not impinge on the analysis of copy raising presented here, which is also base-generated as one would expect in LFG, because the lack of the narrow scope reading is independent of the pronoun. In fact, the lack of this reading must be independent of the pronoun, because the same scope facts hold for physical perception verbs, and we have already observed that these verbs do not require a pronominal copy in their complement. On the present analysis, the lack of the narrow scope reading has to do with the interaction of the standard meaning constructor for a quantifier with the meaning constructor corresponding to the copy raising verb, or the physical perception verb. I will examine this issue in more detail in section 4.3, but it is useful to anticipate the discussion for copy raising verbs, because it highlights a possible relationship between these verbs and control verbs that this theory can capture.

In a copy raising case like (67), the narrow scope reading is blocked because the copy raising verb operates like an anaphoric control verb at the syntax-semantic interface. The f-structure (71) shows that the copy raising sentence in (62) is similar to an extraposition sentence with a raising verb (e.g., "It seems that Richard won."). And the meaning language side of the glue meaning constructor for *seem* in (72) indicates that it has a raising semantics, i.e. it is a one-place predicate taking a propositional argument, just like SSR *seem* with an infinitival complement. However, the linear logic side of the meaning constructor is similar to the right hand side of the Serbo-Croatian anaphoric control verb *pokušao*. In both cases the verb consumes a matrix argument, the subject, and the resource corresponding to its clausal complement after the complement has combined with all arguments save its subject. I have shown elsewhere that the structure of glue proofs guarantees only a wide scope reading for quantified arguments of control verbs (Asudeh 2000, 2002), and this carries over straightforwardly to copy raising. Thus, the scopal behaviour of copy raising verbs reduces to the glue specification of their syntax-semantics mapping, which is shared with control verbs.

This relation between copy raising verbs and control verbs is not accidental. Potsdam and Runner (2002) observe that instances of apparent copy raising with the copied pronominal in non-subject position, as in (68), require that a thematic role be assigned to the matrix subject.

(68) Richard seems like you can rely on him.

In such cases they claim that the verb *seems* means something like "'act like' or 'put on the appearance of'" (Potsdam and Runner 2002). In effect this means that copy raising verbs can sometimes be control verbs.

However, the evidence they provide is not conclusive. First, behavioural evidence of the sort provided by somebody 'acting like' something is often the best grounds for saying they 'seem' like something. The meaning of these two predicates is thus somewhat indistinct. Second, and as a result, it is not clear that the entailments they mention hold. In particular, it does seem possible to conclude from the assertion "He seems like Kim just dumped him" that "It seems that Kim dumped him", contra Potsdam and Runner's (19), repeated here as (69).

(69) a. He seems like Kim just dumped him.
b. =He's acting like Kim just dumped him.
c. ≠It seems that Kim just dumped him.
(Potsdam and Runner 2002: (19))

In other words, I do not share the intuition that their (19c) is not an available reading for their (19a).

Third, the examples they give in these arguments confound raising verbs like *seem* and *appear* with verbs of direct physical perception like *sound*, *look*, *smell*, etc., which have different behaviour, as shown in (64) below. In particular, while the physical perception verbs can lack any pronominal whatsoever in their complement, the raising verbs need a pronoun somewhere in the complement. In what follows, therefore, I do not assume that the copy raising verb necessarily needs a pronoun as the subject of its complement, but only that there is an appropriate pronoun somewhere in the complement.

4.1 Copy Raising

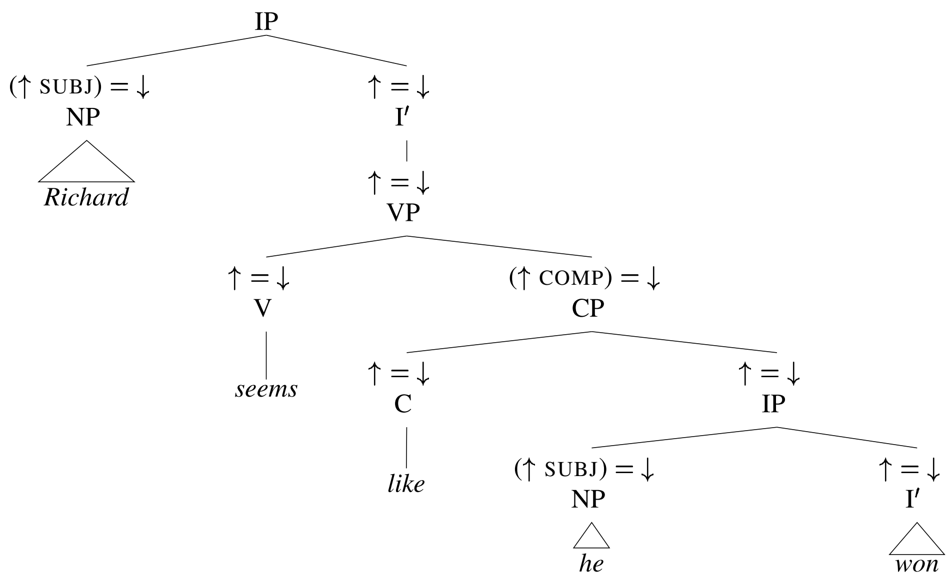
Let us now consider in detail sentence (62), an example with the copy raising verb *seems*.

(62) Richard seems like he won.

Since *seems* is a copy raising verb its subject must bind a pronoun in the complement introduced by *like*.

In assigning a c-structure to (62), the categorical status of most of the constituents is clear, except for the word *like*. Potsdam and Runner (2002) assume, following Heycock (1994) and Maling (1983), that *like* is a preposition. I will not address this issue here, except to note that this assumption sets two generalizations at odds with each other: first, raising verbs like *seems* normally take clausal complements; second, prepositions normally do not take clausal complements. I therefore treat *like* as a complementizer, although I believe the analysis would not change substantially if I were to instead assume that it is a preposition.[13] The c-structure for (62) is

(70)



The lexical entries I am assuming are given in (72). Together with the c-structure above they construct f-structure (71). I have instantiated relevant paths in the lexical entries with node labels from this f-structure.

(71)
$$f\begin{bmatrix} \textsc{pred} & \text{‘seem’} \\ \textsc{subj} & g\begin{bmatrix} \textsc{pred} & \text{‘Richard’} \end{bmatrix} \\ \textsc{comp} & h\begin{bmatrix} \textsc{pred} & \text{‘win’} \\ \textsc{subj} & i\begin{bmatrix} \textsc{pred} & \text{‘pro’} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

[13]There is a notable complication here, caused by sentences like the following:

(i) Richard seems like a nice guy.

In this sentence *like* has a nominal complement, which would fit in with it being a preposition, and *seems* apparently does not have a clausal complement. I have nothing to say about these cases at this point.

(72) *Richard* N $(\uparrow \text{PRED}) = \text{'Richard'}$

$$richard : \uparrow_\sigma$$
$$= g_\sigma$$

seems V $(\uparrow \text{PRED}) = \text{'seem'}$
$(\uparrow \text{COMP GF}^+) = \%\text{CRPronoun}$
$(\%\text{CRPronoun}_\sigma\ \text{ANTECEDENT}) = (\uparrow \text{SUBJ})_\sigma$

$$\lambda x \lambda P.seem(P(x)) : (\uparrow \text{SUBJ})_\sigma \multimap ((\%\text{CRPronoun}_\sigma \multimap (\uparrow \text{COMP})_\sigma) \multimap \uparrow_\sigma)$$
$$= g_\sigma \multimap ((i_\sigma \multimap h_\sigma) \multimap f_\sigma)$$

$$\lambda x \lambda y.x : (\%\text{CRPronoun}_\sigma\ \text{ANTECEDENT}) \multimap (\%\text{CRPronoun}_\sigma \multimap (\%\text{CRPronoun}_\sigma\ \text{ANTECEDENT}))$$
$$= g_\sigma \multimap (i_\sigma \multimap g_\sigma)$$

like C $\lambda I\!P.\ell(I\!P) : \uparrow_\sigma \multimap \uparrow_\sigma$
$$= h_\sigma \multimap h_\sigma$$

he N $(\uparrow \text{PRED}) = \text{'pro'}$

$$\lambda y.y \times y : (\uparrow_\sigma\ \text{ANTECEDENT}) \multimap (\uparrow_\sigma \otimes (\uparrow_\sigma\ \text{ANTECEDENT}))$$
$$= g_\sigma \multimap (i_\sigma \otimes g_\sigma)$$

won V $(\uparrow \text{PRED}) = \text{'win'}$

$$\lambda x.win(x) : (\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma$$
$$= i_\sigma \multimap h_\sigma$$

Most of these entries are of kinds that we have already encountered, but the entries for *seems* and *like* need some further comment.

First, let us consider *like*. It is a complementizer that acts semantically as a modifier on the clause that it introduces, akin to a sentential adverb. *Like* takes the proposition that is the meaning of the clause and applies a modal operator to it. I have indicated this operator as 'ℓ'. I will not define this operator for the moment, but I have in mind something like a belief operator or possibly an evidential. For now, let us just take ℓ as a place-holder.

Next let us turn to *seems*. The second line in this entry features two pieces of notation that we have not seen yet. The path ($\uparrow$ COMP GF$^+$) uses Kleene plus to pick out some grammatical function arbitrarily deeply embedded in the COMP of *seems*. The subject of the complement of *seems*, ($\uparrow$ COMP SUBJ), is picked out by this path, but so is the complement's object, ($\uparrow$ COMP OBJ), the complement's complement's subject, ($\uparrow$ COMP COMP SUBJ), the complement's infinitival complement's oblique object, ($\uparrow$ COMP XCOMP OBL), and so on. These cases are respectively illustrated in (62) and (73a) to (73c).

(73) a. Richard seems like Emma berated him.

b. Richard seems like Emma said that he is fat.

c. Richard seems like Emma tried to send a stinky cheese sandwich to him again.

The second bit of notation has to do with the value of the ($\uparrow$ COMP GF$^+$) path, %CRPronoun, which is a mnemonic for "copy raising pronoun". The prefix '%' indicates that this is a *local name* (Kaplan and Maxwell 1996). A local name is used to name an f-structure within a lexical entry or a phrase structure rule element so that the same exact f-structure can be referred to again (Dalrymple 2001: 107–108). Thus, whatever the value of ($\uparrow$ COMP GF^{+}) is, we are going to call it %CRPronoun.

The third line of the entry is a binding equation that states that the antecedent for the pronoun %CRPronoun is the subject of the copy raising verb. This is analogous to the anaphoric control verb *pokušao* ('try') that we examined in section 3, which stated that its complement's subject's antecedent was its own subject, via the equation $((\uparrow \text{COMP SUBJ})_{\sigma}\ \text{ANTECEDENT}) = (\uparrow \text{SUBJ})_{\sigma}$.

The next line has the first meaning constructor that *seems* contributes. On the left hand side, in the meaning language, we see that *seems* is takes a property ($\langle e, t\rangle$-type) and an individual (e-type) to yield one place predicate with a propositional (i.e., t-type) argument. In this respect, it is like SSR *seems*. In other words, copy raising *seems* and subject-to-subject raising *seems* share the same semantics. However, unlike SSR *seems*. copy raising *seems* composes the proposition by independently picking up a function (i.e., property) and an argument and applying the function to the argument. This is reflected on the right hand side of the meaning constructor, in the linear logic, by two implications. Again, this is analogous to an anaphoric control verb: the copy raising verb consumes the resource corresponding to its subject (g_{σ} above) and then consumes an implicational resource. In the case of an anaphoric control verb, this implicational resource corresponds to the control complement after it has found all of its arguments except its subject, the control target. For example, if a control verb has a control complement headed by a transitive verb that contributes a resource $a \multimap (b \multimap c)$, where a is its object's resource and b is its subject's resource, then the implication $b \multimap c$ is the one that the control verb consumes. The copy raising case is more general, because as we have seen the copy pronoun is not necessarily the subject of the complement, or even necessarily an argument of the complement itself. However, no matter how deeply embedded the copy pronoun is, through an application of implication introduction, we can create a dependency between the copy pronoun and the complement of the copy raising verb. It is this dependency that is consumed by the copy raising verb, just like the anaphoric control verb consumes the dependency from its control target to its controlled complement.

For example, let us suppose that the copy pronoun is the oblique object of the complement of a control verb and that the control verb is the head of the copy raising verb's complement, as in (73c) above. Let us suppose that we have the following meaning constructors for the material in the copy raising verb's complement, leaving aside the meaning language side of the meaning constructors, the modifier *again*, and the pronoun *him*:

(74)	*Emma*	a
	try	$a \multimap ((a \multimap c) \multimap b)$
	send	$pro \multimap (e \multimap (a \multimap c))$
	a stinky cheese sandwich	$\forall X.(e \multimap X) \multimap X$

What we want is an implication of the form $(pro \multimap b)$, because this is what *seems* needs to consume. Here is how we get it:

(75)

$$\dfrac{\dfrac{\dfrac{a \qquad a \multimap ((a \multimap c) \multimap b)}{(a \multimap c) \multimap b}\multimap_{\mathcal{E}} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[pro]^1 \qquad pro \multimap (e \multimap (a \multimap c))}{e \multimap (a \multimap c)}\multimap_{\mathcal{E}} \qquad [e]^2}{a \multimap c}\multimap_{\mathcal{E}} \qquad [a]^3}{c}\multimap_{\mathcal{E}}}{e \multimap c}\multimap_{\mathcal{I},2} \qquad \forall X.(e \multimap X) \multimap X}{c}\multimap_{\mathcal{E}}}{a \multimap c}\multimap_{\mathcal{I},3}}{}}{b}}{pro \multimap b}\multimap_{\mathcal{I},1}$$

No matter how deeply embedded the copy pronoun is, we can use implication introduction to form an implication from the pronoun to the complement of the copy raising verb.

Let us briefly review what the copy raising verb does. It looks arbitrarily deeply in its complement for some pronoun, the copy pronoun. The verb states that its subject is the antecedent of this pronoun. The copy raising verb consumes the resource corresponding to its subject; it also consumes its complement, a dependency on copy pronoun (i.e., an implicational resource from the copy pronoun meaning to the complement meaning). In doing so the copy raising verb gives its subject as an argument to the function corresponding to its complement. The final result is that the copy raising verb is a one-place predicate with a propositional argument.

It will have become obvious by now that I am drawing quite a direct analogy between copy raising verbs and anaphoric control verbs on the one hand and subject raising verbs on the other. Copy raising verbs are like subject raising verbs in their semantics: the proposition corresponding to the raising verb's complement is a semantic argument of the raising verb, but the raising verb's subject is not. The resemblance to anaphoric control verbs is found in the syntax and the syntax-semantics interface. Unlike subject raising verbs, copy raising verbs have a finite complement, and there is no possibility of sharing an argument by functional control. However, there is a necessary anaphoric link between the copy raising subject and some grammatical function in its complement. Syntactically, this anaphoric link is established as anaphoric links standardly are: by using a pronoun. Anaphoric control verbs also establish an anaphoric link by having a pronoun in their complement (the complement's subject) that is anaphoric on a grammatical function of the control verb (the subject in the Serbo-Croatian case we looked at).

The crucial similarity between anaphoric control verbs and copy raising verbs lies at the syntax-semantics interface, though. This brings us to the final matter of the second meaning constructor provided by the copy raising verb. This is a manager resource that consumes the copy pronoun, just like an anaphoric control verb has a manager resource that consumes the controlled subject. In the latter case, the pronoun is there purely as an anaphoric device. It is not serving as a true argument to the clause that contains it, because this clause is not contributing a propositional semantics, but is rather contributing a property. However, if we were to abandon the property semantics for the controlled complement and instead adopt a propositional semantics for it,[14] the pronoun consumption would become unnecessary, as the pronoun would contribute to the propositional semantics of its clause.

In the copy raising case, we really only want the raised subject (not the copy pronoun) to contribute to the semantics, but to the semantics of the clause in which we find the copy pronoun. We accomplish this by consuming the raised subject upstairs, and applying a function with a hole corresponding to the pronoun to the raised subject's meaning. But this means that we must manage the resource of the copy pronoun, or else we would end up with an extra resource and the glue derivation would fail. This is why the copy raising verb has a second meaning constructor, a manager resource that consumes the copy pronoun.

We can now turn to the glue derivation for (62):

(76)

$$\dfrac{\dfrac{\dfrac{\lambda I\!P.\ell(I\!P) : h \multimap h \qquad \dfrac{[v : i]^1 \qquad \lambda x.win(x) : i \multimap h}{win(v) : h}}{\ell(win(v)) : h}}{\lambda v.\ell(win(v)) : i \multimap h}\multimap_{\mathcal{I},1} \qquad \dfrac{\lambda x\lambda P.seem(P(x)) : g \multimap ((i \multimap h) \multimap f) \qquad \dfrac{\dfrac{\dfrac{richard : g \quad \lambda y.y \times y : g \multimap (i \otimes g)}{richard : i,\ \ richard : g}\multimap_{\mathcal{E}},\otimes_{\mathcal{E}} \qquad \lambda x\lambda y.x : g \multimap (i \multimap g)}{richard : i,\ \ \lambda y.x : i \multimap g}\multimap_{\mathcal{E}}}{richard : g}\multimap_{\mathcal{E}}}{\lambda P.seem(P(richard)) : (i \multimap h) \multimap f}\multimap_{\mathcal{E}}}{seem(\ell(win(richard))) : f}\multimap_{\mathcal{E}}$$

On the top right, in the first line, the copy pronoun picks up its antecedent, $richard : g_\sigma$. In the two following lines, the manager resource contributed by the copy raising verb *seems* consumes the pronominal resource, leaving the resource corresponding to the copy raising subject, $richard : g_\sigma$. This is what we already observed in proof (60) for anaphoric control. As in that case, the rest of the derivation proceeds as if the pronoun had never been there. On the fourth line of the right hand side, $richard : g_\sigma$ combines with the other meaning constructor contributed by *seems*, $\lambda x\lambda P.seem(P(x)) : g_\sigma \multimap ((i_\sigma \multimap h_\sigma) \multimap f_\sigma)$. The implication reduction on the linear logic side corresponds

[14]If we were to do this, there would still be the matter of Chierchia's (1984a, 1984b) arguments to counter.

to function application on the meaning side, yielding $\lambda P.seem(P(richard)) : (i \multimap h) \multimap f$ as the result, on the right side of the second to last line of the derivation.

Turning to the left hand side of the proof, on the top line we make an assumption, $[v : i]^1$, and combine it with the meaning constructor contributed by the head of the copy raising verb's complement, $\lambda x.win(x) : i \multimap h$. This yields the proposition $win(v) : h$, which *like* modifies to give $\ell(win(v)) : h$. Next we discharge assumption 1, yielding $\lambda v.\ell(win(v)) : i \multimap h$ on the fourth line of the left side. This is given as an argument to the function $\lambda P.seem(P(richard)) : (i \multimap h) \multimap f$ on the right side, yielding $seem(\lambda v.\ell(win(v))(richard)) : (i \multimap h) \multimap f$, which β-reduces to the conclusion, $seem(\ell(win(richard))) : f$.

4.2 Physical Perception Verbs

Physical perception verbs are clearly somehow related to copy raising verbs, as shown in (64) above. The most telling difference is that physical perception verbs do not require a copy pronoun in the complement, as shown by the wellformedness of (66b), which is repeated here as (77).

(77) Richard looks like Emma won.

There is no sense in which *Richard* in (77) could have been "raised" from the complement of *looks*, because there is no position it could occupy, not even one that is otherwise occupied by a pronoun. As a corollary, if *Richard* in no sense belongs to the lower clause, it must somehow be licensed in the upper clause. Since the matrix subject is obviously not an expletive, as it is a referring expression, it must be an argument of *looks*. In terms of our resource-sensitive theory of semantics, this means that since *Richard* is contributing a resource, there must be a consumer for this resource, and the only likely candidate is *looks*.[15]

Thus another important difference between copy raising and physical perception verbs is that although the subject of a copy raising verb is not a semantic argument of the verb, the subject of a physical perception verb is a semantic argument of the verb. Otherwise the two kinds of verbs seem quite similar. In particular, there does not seem to be any motivation for treating the *like* complement of physical perception verbs differently from that of copy raising verbs. We can schematize the semantic difference between a copy raising exemplar, *seems*, and a physical perception exemplar, *looks*, as follows:

(78) a. $\lambda x \lambda P.seem(P(x))$
e.g., $seem(\ell(win(richard)))$

b. $\lambda x \lambda \mathbb{R}.look(x, \mathbb{R})$
e.g., $look(richard, \ell(win(emma)))$

As we have seen, the copy raising verb is a one place predicate taking a propositional argument that it puts together by feeding the copy raising subject to a function that is derived by abstracting over the copy pronoun in the copy raising verb's complement. The physical perception verb does not put together a proposition out of a function and an argument. A physical perception verb's subject is a semantic argument of the verb itself and the verb's complement does not necessarily contain a pronoun. Even if the complement does contain a pronoun bound by the subject, this is not a copy pronoun and must contribute to the normal semantics of its clause. Thus, a physical perception verb takes an *e*-type subject and a *t*-type complement, whereas a copy raising verb just takes a *t*-type complement.

Let us consider example (77), to make matters clearer. The c-structure for this example will be identical to the c-structure in (70) for the copy raising example (62), except that *looks* is substituted for *seems* and *Emma* is substituted for *he*. Unsurprisingly, the resulting f-structure is also very similar. I will not show the c-structure for (77) here, but the f-structure is

[15] In GB/MP, *Richard* must be assigned a theta role by *looks*, otherwise it would be in violation of the Theta Criterion.

(79)
$$f\left[\begin{array}{ll}\text{PRED} & \text{`look'}\\ \text{SUBJ} & g\left[\begin{array}{ll}\text{PRED} & \text{`Richard'}\end{array}\right]\\ \text{COMP} & h\left[\begin{array}{ll}\text{PRED} & \text{`win'}\\ \text{SUBJ} & i\left[\begin{array}{ll}\text{PRED} & \text{`Emma'}\end{array}\right]\end{array}\right]\end{array}\right]$$

The node labels from this f-structure instantiate the path descriptions in the lexical entries for (77).

The lexical entries for (77) are the same as those for (62) in (72), except for the entry for the physical perception verb *looks* and the name *Emma*, which follow:

(80) *looks* V $(\uparrow \text{PRED}) = \text{`look'}$

$$\lambda x \lambda \mathbb{R}.look(x,\ \mathbb{R}) : \quad (\uparrow \text{SUBJ})_\sigma \multimap ((\uparrow \text{COMP})_\sigma \multimap \uparrow_\sigma)$$
$$= g_\sigma \multimap (h_\sigma \multimap f_\sigma)$$

Emma N $(\uparrow \text{PRED})$

$$emma : \quad \uparrow_\sigma$$
$$= i_\sigma$$

We can construct the following rather straightforward derivation:

(81)
$$\dfrac{\dfrac{\dfrac{emma : i \qquad \lambda x.win(x) : i \multimap h}{win(emma) : h} \qquad \lambda \mathbb{P}.\ell(\mathbb{P}) : h \multimap h}{\ell(win(emma)) : h} \qquad \dfrac{richard : g \qquad \lambda x \lambda \mathbb{R}.look(x,\ \mathbb{R}) : g \multimap (h \multimap f)}{\lambda \mathbb{R}.look(richard,\ \mathbb{R}) : h \multimap f}}{look(richard,\ \ell(win(emma))) : f}$$

On the left hand side we construct the proposition $\ell(win(emma)) : h$ by two applications of modus ponens. On the right hand side, *looks* takes $richard : g$ as its first argument, and then takes the result of the left hand side as its second, yielding the conclusion $look(richard,\ \ell(win(emma))) : f$.

Although a physical perception verb does not require a copy pronoun in its complement, unlike a copy raising verb, we must deal with examples (82), where there just happens to be a pronoun in the complement.

(82) Richard looks like he won.

This results in an f-structure similar to (79):

(83)
$$f\left[\begin{array}{ll}\text{PRED} & \text{`look'}\\ \text{SUBJ} & g\left[\begin{array}{ll}\text{PRED} & \text{`Richard'}\end{array}\right]\\ \text{COMP} & h\left[\begin{array}{ll}\text{PRED} & \text{`win'}\\ \text{SUBJ} & i\left[\begin{array}{ll}\text{PRED} & \text{`pro'}\end{array}\right]\end{array}\right]\end{array}\right]$$

The proof is similar to the one above, except that the pronoun must pick up its antecedent.

(84)
$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[v : i]^1 \qquad \lambda x.win(x) : i \multimap h}{win(v) : h} \qquad \lambda \mathbb{P}.\ell(\mathbb{P}) : h \multimap h}{\ell(win(v)) : h} \qquad \dfrac{[u : g]^2 \qquad \lambda x \lambda \mathbb{R}.look(x,\ \mathbb{R}) : g \multimap (h \multimap f)}{\lambda \mathbb{R}.look(u,\ \mathbb{R}) : h \multimap f}}{look(u,\ \ell(win(v))) : f} \qquad \dfrac{richard : g \qquad \lambda y.y \times y : g \multimap (i \otimes g)}{richard \times richard : (i \otimes g)}}{let\ richard \times richard\ be\ u \times v\ in\ look(u,\ \ell(win(v))) : f}\otimes_{\mathcal{E},1,2}}{look(richard,\ \ell(win(richard))) : f}$$

The left hand side of the proof is essentially proof (81), except that we have assumed $[v : i]^1$ and $[u : g]^2$, rather than using $emma : i$ and $richard : g$ (the former of which is not even in the premise set). On the right hand side, the pronoun picks up its antecedent, $richard : g$, yielding $richard \times richard : (i \otimes g)$. The result of this anaphoric resolution is then combined with the result of the left side, $look(u,\ \ell(win(v))) : f$, by an application of conjunction elimination. The final result reduces to $look(richard,\ \ell(win(richard))) : f$, as we would expect.

4.3 Consequences of the Resource Management Approach to Copy Raising and Physical Perception Verbs

4.3.1 Quantifier and Modifier Scope

Potsdam and Runner (2002), following Lappin (1984), discuss a contrast between subject-to-subject raising verbs and copy raising verbs with respect to quantifier scope. Example (67) above illustrated that a copy raising verb cannot take scope over a quantified subject:

(67) a. Two people seem like they have won the lottery. *seem > 2, 2 > seem

b. ≠ It seems like two people have won the lottery. seem > 2

c. = Two people are such that they seem like they have won the lottery. 2 > seem

(Potsdam and Runner 2002: (35))

This contrasts with the subject-to-subject raising example (85), where the raising verb can take scope over the quantified subject.

(85) a. Two people$_i$ seem t$_i$ to have won the lottery. seem > 2, 2 > seem

b. = It seems that two people have won the lottery. seem > 2

c. = Two people are such that they seem to have won the lottery. 2 > seem

(Potsdam and Runner 2002: (34))

The reason for the scope differences are intuitively clear. An SSR verb can take scope over its subject because its subject is also the subject of the SSR verb's complement. However, the copy raising verb cannot take scope over its subject because verbs in general do not take scope over their subjects and the copy raising subject does not originate in the lower clause on a base-generated account such as this one for Potsdam and Runner's.

On the present account, the scope differences follow from the glue language, i.e., the syntax-semantics interface. Let us compare two simple sentences in more detail to see why this should be so:

(86) An elf seemed to have screamed.

(87) An elf seemed like he had screamed.

A simplified f-structure for the SSR example (86) is given in (88) and one or the copy raising example (87) is given in (89).

(88)
$$f\begin{bmatrix} \text{PRED} & \text{`seem'} \\ \text{SUBJ} & g\begin{bmatrix}\text{PRED} & \text{`elf'}\\ \text{SPEC} & \begin{bmatrix}\text{PRED} & \text{`a'}\end{bmatrix}\end{bmatrix} \\ \text{XCOMP} & h\begin{bmatrix}\text{PRED} & \text{`scream'} \\ \text{SUBJ} & \end{bmatrix}\end{bmatrix}$$

(89)
$$f\left[\begin{array}{ll} \textsc{pred} & \text{`seem'} \\ \textsc{subj} & g\left[\begin{array}{ll} \textsc{pred} & \text{`elf'} \\ \textsc{spec} & \left[\begin{array}{ll} \textsc{pred} & \text{`a'} \end{array}\right] \end{array}\right] \\ \textsc{comp} & h\left[\begin{array}{ll} \textsc{pred} & \text{`scream'} \\ \textsc{subj} & i\left[\begin{array}{ll} \textsc{pred} & \text{`pro'} \end{array}\right] \end{array}\right] \end{array}\right]$$

The crucial difference between these f-structures is that in the one for the SSR verb there is functional control between the SSR verb's subject and the subject of the SSR verb's complement. In other words, these subjects are identical. The copy raising verb has a pronominal subject in its complement and this pronominal is not identical to the copy raising subject.

This syntactic difference is reflected in the meaning constructors that the two kinds of raising verb contribute. The SSR verb allows its complement to consume the shared subject and then consumes the complement (Asudeh 2000), as in (90). The copy raising verb consumes its subject and then consumes its complement's dependency on the copy pronoun, which is disposed of by the copy raising verb's manager resource. An entry for the copy raising verb is repeated in (91). In both (90) and (91) the feature descriptions are instantiated to the relevant f-structure above.

(90) *seems* V $\lambda I\!\!P.seem(I\!\!P) : \ (\uparrow \textsc{xcomp})_\sigma \multimap \uparrow_\sigma$
$= h_\sigma \multimap f_\sigma$

(91) *seems* V $\lambda x \lambda P.seem(P(x)) : \ (\uparrow \textsc{subj})_\sigma \multimap (((\uparrow \textsc{comp subj})_\sigma \multimap (\uparrow \textsc{comp})_\sigma) \multimap \uparrow_\sigma)$
$= g_\sigma \multimap ((i_\sigma \multimap h_\sigma) \multimap f_\sigma)$

$\lambda x \lambda y.x : \ (((\uparrow \textsc{comp subj})_\sigma\ \textsc{antecedent}) \multimap$
$((\uparrow \textsc{comp subj})_\sigma \multimap ((\uparrow \textsc{comp subj})_\sigma\ \textsc{antecedent})))$
$= g_\sigma \multimap (i_\sigma \multimap g_\sigma)$

Let us also assume the following meaning constructor for *an elf*:

(92) *an elf* NP $\lambda P.a(x, elf(x), P(x)) : \ \forall \mathrm{G}.[((\textsc{spec}\uparrow)_\sigma \multimap \mathrm{G}) \multimap \mathrm{G}]$
$= \forall \mathrm{G}.[(g_\sigma \multimap \mathrm{G}) \multimap \mathrm{G}]$

The meaning constructor for *screamed* will be instantiated to $g_\sigma \multimap h$ for (86)/(88) and to $i_\sigma \multimap h_\sigma$ for (87)/(89). This difference will be crucial in accounting for the different scope possibilities:

(93) *screamed* V $\lambda x.scream(x) : \ (\uparrow \textsc{subj})_\sigma \multimap \uparrow_\sigma$
$= g_\sigma \multimap h_\sigma$ for (86)/(88)
$i_\sigma \multimap h_\sigma$ for (87)/(89)

Lastly, let us simplify matters by leaving aside the meaning constructors for the auxiliary verbs in (86) and (87).

Now we are in a position to see how the two scope possibilities for the SSR example (86) can arise. In the first case, the verb takes wide scope over its subject, because the subject takes the SSR verb's complement as its scope (i.e., its scope position is in the XCOMP). I leave out the meaning language, save on the conclusion, for simplicity's sake. The meaning language follows straightforwardly from the Curry-Howard isomorphism.

(94)
$$\dfrac{\dfrac{g \multimap h \qquad \forall G.[(g \multimap G) \multimap G]}{h}\,G = h \qquad\qquad h \multimap f}{seem(a(x, elf(x), scream(x))) : f}$$

Alternatively, we can make an assumption on g, feed it to the complement, deriving a proposition. We can then feed this proposition to the raising verb, discharge the assumption and scope the quantifier in:

(95)

$$\dfrac{\dfrac{\dfrac{\dfrac{[g]^1 \qquad g \multimap h}{h} \qquad h \multimap f}{f}}{g \multimap f}\multimap_{\mathcal{I},1} \qquad \forall G.[(g \multimap G) \multimap G]}{a(x, \mathit{elf}(x), \mathit{seems}(\mathit{scream}(x))) : f}\; G = f$$

The reason both these scopings are possible is that the quantified NP is a semantic argument of the complement, not of the raising verb. In terms of the glue language, the quantifier is a dependency on g_σ and the complement forms a dependency on this s-structure node as well. However, the raising verb is dependent only on the resolved complement, h_σ.

The copy raising verb is quite different, though. In (87), the quantified NP *an elf* is a semantic argument of the copy raising verb, which forms a dependency on it, i.e., $g_\sigma \multimap ((i_\sigma \multimap h_\sigma) \multimap f_\sigma$. In particular, the quantifier cannot be scoped in the complement, because in the complement there is no dependency on g_σ, the s-structure node the quantifier depends on. In other words, in order to scope the quantifier in the complement, we would need to have a resource of the form $g_\sigma \multimap h_\sigma$. However, we do not have this resource in our premise set and there is no way to construct it as a conclusion from any of the premises we do have.

The only way to scope the quantifier in is to resolve the copy raising verb first, using an assumption on g_σ. The copy raising verb can combine with this assumption and then the dependency $i_\sigma \multimap h_\sigma$ that corresponds to its complement, yielding a proposition. We then discharge the assumption and scope in the quantifier:[16]

(96)

$$\dfrac{\dfrac{\dfrac{\dfrac{[g]^1 \qquad g \multimap ((i \multimap h) \multimap f)}{(i \multimap h) \multimap f} \qquad i \multimap h}{f}}{g \multimap f}\multimap_{\mathcal{I},1} \qquad \forall G.[(g \multimap G) \multimap G]}{a(x, \mathit{elf}(x), \mathit{seem}(\mathit{scream}(x))) : f}\; G = f$$

Thus, we see that it follows from the syntactic differences between SSR and copy raising verbs, their differences at the syntax-semantics interface, and our theory of the interface, glue semantics, that an SSR verb can take scope over its quantified subject, while a copy raising verb cannot. Physical perception verbs in this respect are like copy raising verbs: since the quantier is a semantic argument of the physical perception verb and not of its complement, it can only take wide scope over the verb.

Potsdam and Runner (2002) discuss another scope difference, having to do with quantificational adverbs and bare plurals. An adverb in the complement of an SSR verb can take scope over the matrix subject (98), but an adverb in the complement of a copy raising verb cannot (100).

(97) a. Cows rarely seem to be intelligent

b. = Few cows seem to be intelligent

c. rarely$_x$ cow(x) seem [...]

(Potsdam and Runner 2002: (36))

[16] Note that I have assumed that the pronoun has been disposed of by the copy raising verb's manager resource. I have left this step out as we have already seen it above and it would needlessly complicate the point at hand.

(98) a. Cows seem rarely to be intelligent.

b. = It seems that few cows are intelligent.

c. seem [rarely$_x$ cow(x)...]

(Potsdam and Runner 2002: (37))

(99) a. Cows rarely seem like they are intelligent.

b. = Few cows seem like they are intelligent.

c. rarely$_x$ cow(x) seem [...]

(Potsdam and Runner 2002: (38))

(100) a. *Cows seem like they are rarely intelligent.

b. ?It seems like few cows are intelligent.

c. *cow(x) seem [rarely$_x$...]

(Potsdam and Runner 2002: (39))

A sentential adverb like *rarely* will be a modifier on sentential resources. Supposing that the f-structure node label for the sentential complement in (98) is a and its s-structure is a_σ, then *rarely* would contribute a resource $a_\sigma \multimap a_\sigma$.

The SSR subject is simultaneously the subject of the SSR verb and of the complement in (100). It is the complement that picks it up, yielding a proposition. The SSR verb has a glue meaning constructor of the form $a_\sigma \multimap b_\sigma$, where a_σ is the resource corresponding to the SSR verb's sentential complement and b_σ is the resource corresponding to the clause that the SSR verb heads. The SSR subject is picked up by the sentential complement, which can then be modified by *rarely*. The complement is then picked up by the SSR verb. This yields precisely the scoping in (98c).

However, the copy raising subject in (100) is not an argument of the copy raising verb's sentential complement. Thus, when *rarely* modifies the complement, we cannot get a reading in which *rarely* takes scope over the copy raising subject, because this subject is outside the clause that *rarely* modifies. The theory yields the correct scope results for modifiers like quantificational adverbs for the same reason that it yields the correct scope results for quantified subjects; the subject of a copy raising verb is not a semantic argument of its complement and can therefore never take scope a) under the verb, or b) inside the sentential complement of the verb, under modifiers of the complement.

4.3.2 A Typology of Verbs

The analysis of copy raising and physical perception verbs in terms of resource management gives an interesting typology of related verbs:

(101)

Verb class	Syntax	Semantics	Syntax-Semantics	Manager resource?
Physical perception	no control	Proposition	$a \multimap (b \multimap c)$	no
Copy raising	anaphoric	Proposition	$a \multimap ((p \multimap b) \multimap c)$	yes
Raising	functional	Proposition	$b \multimap c$	no
Balkan-type control	anaphoric	Property	$a \multimap ((p \multimap b) \multimap c)$	yes
English-type control	functional	Property	$a \multimap ((a \multimap b) \multimap c)$	no

This table shows that there are four dimensions of variation for these verbs: the syntactic relationship between a matrix and a subordinate argument, the semantics of the complement, the glue language, and the presence of a manager resource. The two kinds of control verb differ on three dimensions, but both kinds have a complement that denotes a property (Chierchia 1984a,b, Zec 1987). Similarly, raising and copy raising verbs share only the semantic dimension. But, there are similarites between raising verbs and functional control verbs on the one hand and between copy raising and anaphoric control verbs on the other. The former pair share the syntactic mechanism of functional control and the absence of a manager resource, while the latter pair share the syntactic mechanism of anaphoric control, and have the same glue language, necessitating the presence of a manager resource. Lastly, raising and copy raising verbs share the semantic dimension with physical perception verbs.

5 Resumptive Pronouns

In this section I present an analysis of resumptive pronouns in Irish, which is illustrative of the general resource management approach to this phenomenon. Irish is especially amenable to the study of resumptive pronouns, because the complementizers in Irish, *go*, *aL*, and *aN* morphologically register extraction activity, including resumption, in the clauses that they introduce (McCloskey 1979, to appear, Sells 1984). Roughly, the complementizer *go* signals the absence of a long distance dependency passing through its clause, *aL* signals the presence of a long distance dependency with no resumptive pronoun at the foot, and *aN* registers a long distance dependency that terminates in a resumptive:

(102) Creidim gu-r inis sé bréag.
believe.1SG go-PAST tell he lie
I believe that he told a lie.
(McCloskey to appear: 7, (8))

(103) an ghirseach a ghoid na síogaí
the girl aL stole the fairies
the girl that the fairies stole away
(McCloskey to appear: 7, (9a))

(104) an ghirseach a-r ghoid na síogaí í
the girl aN-PAST stole the fairies her
the girl that the fairies stole away
(McCloskey to appear: 7, (9b))

It is not sufficient to deal with the resumptive cases alone. It is also necessary to show that the analysis is general and can handle examples with the other complementizers, like (102) and (103). The analysis I present below accounts for the sentences in (102)–(104), as well as various more complex cases discussed by McCloskey (to appear).

I think it would be helpful to anticipate the analysis somewhat, and to present its main ingredients, even if the details do not make sense yet. The wellformedness or illformedness of relevant sentences will be determined by the two classes of criteria in (105).

(105) wellformedness criteria

1. General syntactic wellformedness criteria, including the lexical specifications of items that feed into the syntax
2. Glue semantic derivations, in particular their resource-sensitivity, which will treat derivations with unused resources as illformed and derivations that use each resource exactly once as wellformed

Given these wellformedness criteria, there will be three main ingredients in the LFG analysis I present below:

(106) Main ingredients of the analysis of Irish

1. Lexical entries for the comlementizers *go*, *aL*, and *aN*
2. Annotated phrase-structure rules
3. Glue meaning constructors, in relevant lexical entries and on relevant phrase structure rules

Picking up on the first ingredient — the lexical entries for the complementizers — we will see that there is a theoretically interesting classification to be made. We can define two levels of grammatical activity on which lexical entries operate:

(107)
1. **Syntactic activity**: A lexical item is *syntactically active* if and only if it contributes a functional equation other than $\uparrow = \downarrow$.
2. **Semantic activity**: A lexical item is *semantically active* if and only if it contains a glue meaning constructor (i.e., it contributes a resource to the glue derivation).

Here are the lexical entries assumed in the analysis below. The details are not important yet, just very general facts about these entries.

(108) a. *go* C $\uparrow = \downarrow$

b. *aL* C $\left\{ (\uparrow \textsc{topic}) = (\uparrow \textsc{comp topic}) \;\middle|\; \begin{array}{l} (\uparrow \textsc{topic}) = (\uparrow \textsc{gf}) \\ (\uparrow \textsc{topic pred}) = \text{‘pro’} \\ (\uparrow \textsc{topic prontype}) = \text{variable} \end{array} \right\}$

c. *aN* C $(\uparrow \textsc{topic}) = (\uparrow [\textsc{gf}^{+} - \textsc{subj}])$
$(\uparrow \textsc{topic prontype})$
$((\uparrow \textsc{topic})_\sigma\ \textsc{antecedent}) \multimap ((\uparrow \textsc{topic})_\sigma \multimap ((\uparrow \textsc{topic})_\sigma\ \textsc{antecedent}))$

According to the definitions of syntactic and semantic activity, the following classification emerges:[17]

(109) Classification of the Irish complementizers wrt grammatical activity

	Syntactically Active	Semantically Active
go	no	no
aL	yes	no
aN	yes	yes

Turning to the second ingredient of the analysis, the following annotated phrase structure rules for CP construction will be important:

(110) CP $\longrightarrow$ $\begin{array}{c} \text{C}' \\ \uparrow = \downarrow \\ \left(\begin{array}{c} \textbf{[rel]} \\ ((\uparrow \textsc{topic})_\sigma\ \textsc{antecedent}) = (\textsc{adj} \in \uparrow)_\sigma \end{array} \right) \end{array}$

(111) C′ $\longrightarrow$ $\begin{array}{c} \text{C} \\ \uparrow = \downarrow \\ \left(\begin{array}{l} [(\uparrow \textsc{topic})_\sigma\ \textsc{antecedent}] \multimap \\ \quad [(\uparrow \textsc{topic})_\sigma \otimes ((\uparrow \textsc{topic})_\sigma\ \textsc{antecedent})] \end{array} \right) \end{array}$ $\begin{array}{c} \text{IP} \\ \uparrow = \downarrow \end{array}$

The C′ rule has an optional functional equation which provides a pronominal resource; various realizations of this option will work in conjunction with the lexical entries for the complementizers and information in the IP to account for the distribution of the complementizers.

The CP rule optionally provides a glue meaning constructor for constructing the semantics of a relative clause, following Dalrymple (2001). This optional meaning constructor, abbreviated as **[rel]**, will be realized in relative clauses,[18] but not when the CP is serving as a complement (i.e., it bears the COMP grammatical function). Additionally,

[17] Note that the classification applies to uninflected *base* forms of the complementizers, as they can be inflected for tense, and then inflected versions of *go* would count as syntactically active.

[18] The full definition of **[rel]** will be given in (116) below.

when **[rel]** is realized, a binding equation is provided that sets the ANTECEDENT of the TOPIC of the relative clause to the σ-mapping of the nominal that the relative clause modifies. In other words, this equation states that the antecedent of the resumptive pronoun or missing grammatical function (i.e., trace on a transformational account) in the relative clause is the relative head. The constructor **[rel]**, as well as the meaning constructor provided by the complementizer *aN* and the meaning constructors provided by the other Irish lexemes, are the third ingredient of the analysis. The interplay of these glue resources is the second criterion for wellformedness noted above.

There are three basic intuitions behind this account:

1. There is a one to one correspondence between *aN* and pronominal resources. If there are multiple *aN*s in a sentence, such that there is only one resumptive pronoun, all but one *aN* must be consuming a pronominal resource provided by the C$'$ rule.
2. Thee *aN*-pronoun relationship is unbounded, whereas the *aL*-gap relationship is locally bounded and cyclic.
3. Syntax and semantics place separate requirements on wellformedness and there can be complex interactions between these requirements.

The layout of this section is as follows. In section 5.2, I account for the distribution of *go* and *aL*, as exemplified by (102) and (103). McCloskey (to appear) also presents a variety of interesting "mixed" cases, which consist of larger relative clauses containing embedded clauses. These cases have multiple complementizers, appearing in various permissible orders. I consider these more complex facts in section 5.3. However, since the central topic of this investigation is resumptives, we will begin by examining *aN* when it introduces a clause containing a resumptive pronoun, as in example (104).

5.1 *aN*

Let us consider the example sentence (104):

(104) an ghirseach a-r ghoid na síogaí í
the girl aN-PAST stole the fairies her
the girl that the fairies stole away

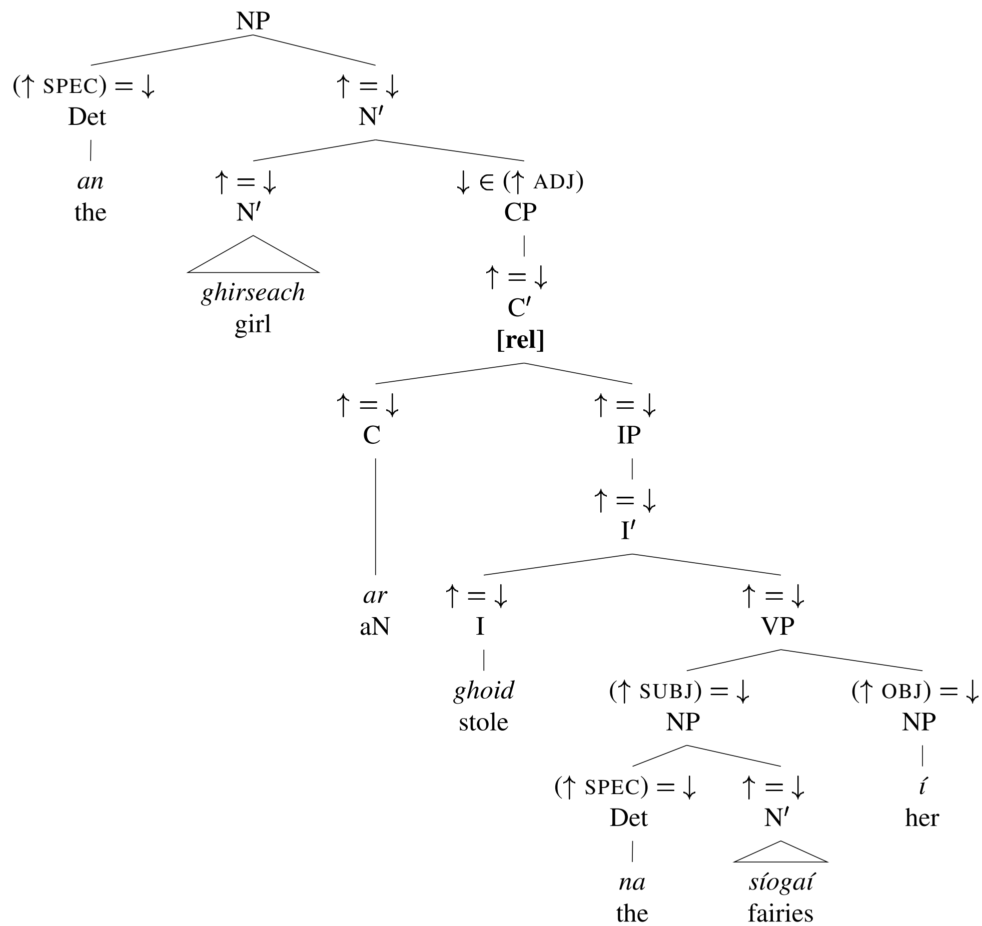
I will assume the c-structure (113) for (104). The phrase structure rule in (112) that constructs the relative clause CP is rule (110) with the optional material realized.

(112) $$\mathrm{CP} \longrightarrow \begin{array}{c} \mathrm{C}' \\ \uparrow = \downarrow \\ \textbf{[rel]} \\ ((\uparrow \text{TOPIC})_\sigma\ \text{ANTECEDENT}) = (\text{ADJ} \in \uparrow)_\sigma \end{array}$$

The CP rule (112) does three things. First, by the $\uparrow = \downarrow$ equation it identifies its f-structural information with that of its mother. Second, the rule provides the glue meaning constructor **[rel]**. If the CP in question is a relative clause, like in (104), then **[rel]** must be provided to construct the relative clause's semantics. Without this meaning constructor, the relative clause semantics will not be composed with that of the N$'$ it modifies. Third, the rule sets the ANTECEDENT of the resumptive pronoun to the σ-structure of the head of the relative clause. In other words, the head of the relative clause is the antecedent of the resumptive pronoun.

With these rules in hand, and assuming the usual rules for the other constituents, we have the following c-structure for (113):

(113)



The lexical entries in (114) are the terminals of this c-structure. The verb *ghoid* ('steal') is a normal transitive verb: it wants to consume its subject's meaning and its object's meaning to yield its meaning. The pronoun *í* ('her') is also just a normal pronoun, seeking to consume its antecedent's meaning to yield a copy of the antecedent meaning and its own meaning. As the definite determiners are not the key aspect of the example, I simplify matters by giving them a Russellian, quantificational semantics, rather than a Heim-Kamp semantics. The determiners each take their N′ as their restriction. The nuclear scope depends on the resource corresponding to the head of the restriction, as usual. For example, *an* ('the') takes *ghirseach* ('girl') as its restriction by consuming the resource corresponding to *ghirseach*, $(f_\sigma \text{ VAR}) \multimap (f_\sigma \text{ RESTR})$. This results in a quantifier that has found its restriction, but is still looking for its scope, a dependency on f_σ, as specified by the resource $\forall G.[(f_\sigma \multimap G) \multimap G]$.

The lexical entry for the complementizer *ar*, a form of the *aN* complementizer which is sensitive to resumptive pronouns, is the key to the analysis. Before considering its glue semantics, let us consider its f-structural information, given in three functional equations at the beginning of the entry. First, it specifies that the clause it introduces is in the past tense, as per the *-r* inflection (see (104) above). I assume that the inflection adds this equation in the morphology, with the base form of *aN* being specified as in (108c) above, with no tense information. Second, the entry requires that its TOPIC have a PRONTYPE;[19] in other words, the TOPIC must be a pronominal. Third, it requires that its TOPIC be any of the grammatical functions of its f-structure, except the SUBJECT,[20] or a grammatical function

[19]The feature PRONTYPE is used to distinguish different kinds of pronominals, such as personal pronouns, relative pronouns, wh-pronouns, etc., subject to the distinctions made by the language under analysis (Butt et al. 1999, Dalrymple 2001).

[20]This encodes the highest subject restriction: in Irish the highest relative clause subject cannot be a resumptive (McCloskey to appear).

of some f-structure that is arbitrarily embedded in the complementizer's f-structure. In other words, *aN* identifies its TOPIC, i.e., the grammatical function that is relativized, by looking for a pronoun in either its f-structure, or some f-structure embedded within its f-structure. Like pronominal relations in general, this is an unbounded search. How is the correct pronoun, i.e., the resumptive, identified? Recall that the CP rule in (112) that constructs the relative clause also provides a functional equation that sets the antecedent of the TOPIC, in this case the resumptive pronoun, to be the relative clause's head.

However, independently of this equation, the resource-sensitivity of glue semantics plays a role. The complementizer *aN* (and the inflected version *ar* we are considering here) provides a glue meaning constructor that consumes whatever pronoun is identified with the TOPIC. If the pronoun that is identified with the TOPIC is not a resumptive pronoun, but rather a true argument of some predicate that needs to consume the pronoun's resource, then the linear logic proof will not meet its wellformedness criteria for two reasons. First, the resumptive pronoun will be contributing an excess resource that is not being consumed (since the complementizer has "accidentally" consumed another resource). Second, the resource requirements of anything that depends on the consumed non-resumptive will not be met. At this point this explanation is no doubt confusing; we will return to the complementizer below. For now, here are the lexical entries we have been discussing, followed by the f-structure they help define.

(114) *an* Det $(\uparrow \text{PRED}) = \text{`the'}$

$$\lambda P \lambda Q.the(x, P(x), Q(x)) : \quad [((\text{SPEC} \uparrow)_\sigma\ \text{VAR}) \multimap ((\text{SPEC} \uparrow)_\sigma\ \text{RESTR})] \multimap \forall G.[((\text{SPEC} \uparrow)_\sigma \multimap G) \multimap G]$$
$$= [(f_\sigma\ \text{VAR}) \multimap (f_\sigma\ \text{RESTR})] \multimap \forall G.[(f_\sigma \multimap G) \multimap G]$$

ghirseach N $(\uparrow \text{PRED}) = \text{`girl'}$

$$\lambda x.girl(x) : \quad (\uparrow_\sigma\ \text{VAR}) \multimap (\uparrow_\sigma\ \text{RESTR})$$
$$= (f_\sigma\ \text{VAR}) \multimap (f_\sigma\ \text{RESTR})$$

ar C $(\uparrow \text{TENSE}) = \text{past}$
$(\uparrow \text{TOPIC PRONTYPE})$
$(\uparrow \text{TOPIC}) = (\uparrow [\text{GF}^+ - \text{SUBJ}])$

$$\lambda x \lambda y.x : \quad ((\uparrow \text{TOPIC})_\sigma\ \text{ANTECEDENT}) \multimap ((\uparrow \text{TOPIC})_\sigma \multimap ((\uparrow \text{TOPIC})_\sigma\ \text{ANTECEDENT}))$$
$$= f_\sigma \multimap (i_\sigma \multimap f_\sigma)$$

ghoid V $(\uparrow \text{PRED}) = \text{`steal'}$

$$\lambda y \lambda x.steal(x, y) : \quad (\uparrow \text{OBJ})_\sigma \multimap ((\uparrow \text{SUBJ})_\sigma \multimap \uparrow_\sigma)$$
$$= i_\sigma \multimap (h_\sigma \multimap g_\sigma)$$

na Det $(\uparrow \text{PRED}) = \text{`the'}$

$$\lambda P \lambda Q.the(x, P(x), Q(x)) : \quad [((\text{SPEC} \uparrow)_\sigma\ \text{VAR}) \multimap ((\text{SPEC} \uparrow)_\sigma\ \text{RESTR})] \multimap \forall H.[((\text{SPEC} \uparrow)_\sigma \multimap H) \multimap H]$$
$$= [(h_\sigma\ \text{VAR}) \multimap (h_\sigma\ \text{RESTR})] \multimap \forall H.[(h_\sigma \multimap H) \multimap H]$$

síogaí N $(\uparrow \text{PRED}) = \text{`fairy'}$

$$\lambda x.fairy(x) : \quad (\uparrow_\sigma\ \text{VAR}) \multimap (\uparrow_\sigma\ \text{RESTR})$$
$$= (h_\sigma\ \text{VAR}) \multimap (h_\sigma\ \text{RESTR})$$

í N $(\uparrow \text{PRED}) = \text{`pro'}$
$(\uparrow \text{PRONTYPE}) = \text{personal}$
$$\lambda y.y \times y : \quad (\uparrow_\sigma\ \text{ANTECEDENT}) \multimap (\uparrow_\sigma \otimes (\uparrow_\sigma\ \text{ANTECEDENT}))$$
$$= f_\sigma \multimap (i_\sigma \otimes f_\sigma)$$

From these lexical entries and the c-structure above, we get the f-structure in (115). The relative clause is an ADJUNCT of the head of the relative clause, *an ghirseach* (‘the girl’). The OBJ of *ghoid* (‘steal’) is the resumptive pronoun, *í* (‘her’).

(115)
$$f\begin{bmatrix} \text{PRED} & \text{`girl'} \\ \text{NUM} & \text{sg} \\ \text{SPEC} & \begin{bmatrix}\text{PRED} & \text{`the'}\end{bmatrix} \\ \text{ADJ} & \left\{ g\begin{bmatrix} \text{PRED} & \text{`steal'} \\ \text{SUBJ} & h\begin{bmatrix} \text{PRED} & \text{`fairy'} \\ \text{NUM} & \text{pl} \\ \text{SPEC} & \begin{bmatrix}\text{PRED} & \text{`the'}\end{bmatrix} \end{bmatrix} \\ \text{OBJ} & i\begin{bmatrix} \text{PRED} & \text{`pro'} \\ \text{PRONTYPE} & \text{pers} \end{bmatrix} \\ \text{TOPIC} & \\ \text{TENSE} & \text{past} \end{bmatrix} \right\} \end{bmatrix}$$

The rule responsible for constructing the relative clause CP was given in (112) and is repeated here:

(112)
$$\text{CP} \longrightarrow \begin{array}{c} \text{C}' \\ \uparrow = \downarrow \\ \textbf{[rel]} \\ ((\uparrow \text{TOPIC})_\sigma\ \text{ANTECEDENT}) = (\text{ADJ} \in \uparrow)_\sigma \end{array}$$

In addition to setting the antecedent of the resumptive pronoun, as discussed above, the rule provides a meaning constructor, abbreviated as **[rel]**:

(116)
$$\begin{array}{rl} \textbf{[rel]}\ \lambda P \lambda Q \lambda x. P(x) \wedge Q(x) : & [(\uparrow \text{TOPIC})_\sigma \multimap \uparrow_\sigma\,] \multimap \\ & [[((\text{ADJ} \in \uparrow)_\sigma\ \text{VAR}) \multimap ((\text{ADJ} \in \uparrow)_\sigma\ \text{RESTR})] \multimap \\ & \ [((\text{ADJ} \in \uparrow)_\sigma\ \text{VAR}) \multimap ((\text{ADJ} \in \uparrow)_\sigma\ \text{RESTR})]] \\ = & (i_\sigma \multimap g_\sigma) \multimap \\ & [((f_\sigma\ \text{VAR}) \multimap (f_\sigma\ \text{RESTR})) \multimap ((f_\sigma\ \text{VAR}) \multimap (f_\sigma\ \text{RESTR}))] \end{array}$$

This meaning constructor is responsible for composing the relative clause semantics with that of the N$'$ being modified.

Recall that the rule for constructing the C$'$ has an optional annotation:

(111)
$$\text{C}' \longrightarrow \begin{array}{c} \text{C} \\ \uparrow = \downarrow \\ \left(\begin{array}{l} [(\uparrow \text{TOPIC})_\sigma\ \text{ANTECEDENT}] \multimap \\ \quad [(\uparrow \text{TOPIC})_\sigma \otimes ((\uparrow \text{TOPIC})_\sigma\ \text{ANTECEDENT})] \end{array} \right) \end{array} \quad \begin{array}{c} \text{IP} \\ \uparrow = \downarrow \\ \\ \end{array}$$

The optional annotation introduces a pronominal meaning constructor (compare this meaning constructor to the one given in the lexical entry for *í* ('her') in (114) above). For the particular case we are looking at — the complementizer *aN* introducing a clause that contains a resumptive pronoun — the optional equation should *not* be realized. This equation will be discussed in more detail below.

Let us see how meaning composition proceeds in this case. We will look at the proofs without meaning terms, as the meanings follow straightforwardly, according to the Curry-Howard isomorphism (the same proofs with meaning terms specified are provided in appendix A). I present the proofs first, and discuss them below.

(117) Construction of the relative clause semantics
$(f_\sigma\ \text{VAR}) = v$, $(f_\sigma\ \text{RESTR}) = r$

$$\dfrac{v \multimap r \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[i_\sigma]^1 \quad i_\sigma \multimap (h_\sigma \multimap g_\sigma)}{h_\sigma \multimap g_\sigma}\multimap_{\mathcal{E}} \qquad \forall H.[(h_\sigma \multimap H) \multimap H]}{g_\sigma}\multimap_{\mathcal{E}}, H = g}{i_\sigma \multimap g_\sigma}\multimap_{\mathcal{I},1} \qquad (i_\sigma \multimap g_\sigma) \multimap [(v \multimap r) \multimap (v \multimap r)]}{(v \multimap r) \multimap (v \multimap r)}\multimap_{\mathcal{E}}}{}}{v \multimap r}\multimap_{\mathcal{E}}$$

$$\dfrac{v \multimap r \qquad (v \multimap r) \multimap \forall G.[(f_\sigma \multimap G) \multimap G]}{\forall G.[(f_\sigma \multimap G) \multimap G]}\multimap_{\mathcal{E}}$$

(118) Pronoun consumption

$$\dfrac{\dfrac{\dfrac{[f_\sigma]^2 \quad f_\sigma \multimap (i_\sigma \multimap f_\sigma)}{i_\sigma \multimap f_\sigma}\multimap_{\mathcal{E}} \qquad [i_\sigma]^3}{f_\sigma}\multimap_{\mathcal{E}} \qquad \dfrac{[f_\sigma]^4 \quad f_\sigma \multimap (i_\sigma \otimes f_\sigma)}{i_\sigma \otimes f_\sigma}\multimap_{\mathcal{E}}}{f_\sigma}\otimes_{\mathcal{E},2,3}$$

(119) Putting the two parts together

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[f_\sigma \multimap J]^5 \quad f_\sigma}{J}\multimap_{\mathcal{E}}}{f_\sigma \multimap J}\multimap_{\mathcal{I},4} \qquad \forall G.[(f_\sigma \multimap G) \multimap G]}{J}\multimap_{\mathcal{E}}, G = J}{(f_\sigma \multimap J) \multimap J}\multimap_{\mathcal{I},5}}{\lambda P.the(y, the(x, fairy(x), steal(x,y)) \wedge girl(y), P(y)) : \forall K.[(f_\sigma \multimap K) \multimap K]}\forall_{\mathcal{I}}$$

First the semantics of the relative clause is constructed, in subproof (117). We assume a resource corresponding to the OBJ of *steal* and combine this with the resource for *steal* to derive the scope to be combined with the quantifer *the fairies*. We combine the quantifier with its scope to yield a sentential semantics for the relative clause, and then the assumption of the OBJ resource is discharged, abstracting over the corresponding variable in the sentential meaning. The resulting implication is combined with **[rel]**, yielding a modifier on the resource for *girl*. This is combined with *girl* and the result then provides the restriction for the definite determiner *the*, which takes *girl* (modified by the relative clause) as its restriction.

In the second subproof (118), we have pronoun consumption. The resource that comes from the resumptive pronoun ($f_\sigma \multimap [i_\sigma \otimes f]$) is combined with the manager resource corresponding to the complementizer *aN* ($f \multimap [i \multimap f]$). The result is that the complementizer consumes the pronominal resource, yielding the resource corresponding to the outer f-structure, the head of the relative clause.

In the last subproof (119), we assume a resource corresponding to a clause that depends on the NP resource f_σ. By combining the result of the previous subproof with this assumption, we get a sentential resource, and then we discharge the assumption on f from the righthand side of (118), line 2. Now we have an implication resource that corresponds to what the quantifier *the girl* is looking for as its scope. We combine this with the quantifier resource to get the sentential resource J, discharge the assumed $f_{\sigma} \multimap J$ to derive the implication in the second to last line, and finish the proof by performing Universal Introduction[21] in the linear logic, substituting J with K and binding K. The corresponding meaning is a quantifier looking for its scope, which corresponds to the meaning of *an ghirseach ar ghoid na síogaí í* ('the girl the fairies stole her'), which is a quantified NP, with a relative clause modifier, that requires a nuclear scope.

The crucial step was the pronoun consumption in (118): *if the complementizer* aN *were not present, the pronominal resource would not be consumed, and the linear logic proof would fail, as there would be an extra resource.*

We have just seen how the resource management of the complementizer *aN* allows the presence of the resumptive pronoun. We have thus accounted for the pattern that can be schematized as

(120) $[_{\text{CP}}$ aN ... pro]

Here "pro" is a resumptive pronoun, but notice that nothing in the analysis above treated the pronoun as anything other than a regular pronoun. The literature on resumptives is in agreement on the fact that languages do not have distinct resumptive paradigms, as opposed to the normal pronominal paradigms (McCloskey to appear, Sells 1984). In this case the resumptive contributes a normal pronominal PRED, a pronominal glue meaning constructor, and a PRONTYPE appropriate to personal pronouns. The action was in the complementizer *aN*, for which the general entry is repeated here:[22]

(121) *aN* C
$$(\uparrow \text{TOPIC}) = (\uparrow [\text{GF}^{+} - \text{SUBJ}])$$
$$(\uparrow \text{TOPIC PRONTYPE})$$
$$((\uparrow \text{TOPIC})_{\sigma}\ \text{ANTECEDENT}) \multimap ((\uparrow \text{TOPIC})_{\sigma} \multimap ((\uparrow \text{TOPIC})_{\sigma}\ \text{ANTECEDENT}))$$

The complementizer consumes the pronoun's resource and equates the TOPIC in the relative clause with the OBJ grammatical function. The latter is due to the equation in the first line of (121), which requires that the TOPIC of *aN*'s clause be identified with some grammatical function. The GF can be arbitrarily embedded in the f-structure corresponding to *aN* (as indicated by the Kleene plus), but cannot be the highest subject. This satisfies the Extended Coherence Condition (ECC) (Zaenen 1980, Bresnan and Mchombo 1987, Fassi-Fehri 1988, Dalrymple 2001). The ECC requires that TOPIC and FOCUS should be identified with some subcategorized grammatical function either anaphorically (through coindexation) or by identity. The intuition is that TOPICs and FOCI must be integrated into f-structures. And as already discussed, the pronoun consumption satisfies the requirements of the glue semantics, that all resources should be used exactly once.

Recall in addition the C′ rule that was used in constructing the relative caluse:

(111)
$$\text{C}' \longrightarrow \begin{array}{cc} \text{C} & \text{IP} \\ \uparrow = \downarrow & \uparrow = \downarrow \\ \left(\begin{array}{l} [(\uparrow \text{TOPIC})_{\sigma}\ \text{ANTECEDENT}] \multimap \\ \quad [(\uparrow \text{TOPIC})_{\sigma} \otimes ((\uparrow \text{TOPIC})_{\sigma}\ \text{ANTECEDENT})] \end{array} \right) & \end{array}$$

In this case, the optional annotation under C cannot be realized. If it were realized, there would be an extra pronominal resource, leading to glue failure.

What if there were no resumptive below the *aN*? The lexical entry for *aN* and the CP rule together predict the badness of *aN* with no resumptive:

[21] The rule for Universal Introduction is

(i) **Universal Introduction**

$$\frac{\begin{array}{c}\Gamma \\ \vdots \\ P : A\end{array}}{\lambda Q.Q(P) : \forall X.A}\ \forall_{\mathcal{I}},\ X\ \mathit{new\ in}\ \Gamma$$

For more details, see Crouch and van Genabith (2000: 124).

[22] This entry leaves aside the tense-marking of the particular inflected form of *aN* in (104) above.

(122) $*[_{\text{CP}}$ aN . . . t]

If there is no pronominal resource provided by a resumptive pronoun, the only way for *aN*'s manager resource to be satisfied is if the optional annotation on C is chosen. However, in this case nothing is specifying the PRONTYPE of the pronoun, as there is no overt pronoun and no lower dependency introducing a bound variable (see the discussion of mixed chains below). The existential constraining equation on *aN* that checks for PRONTYPE would not be satisfied, as it is checking for the presence of a PRONTYPE feature, and nothing is providing this feature.

Furthermore, nothing would be providing the relativized grammatical function with a PRED value. This violates Completeness (Kaplan and Bresnan 1982, Bresnan 2001, Dalrymple 2001), which requires that all subcategorized grammatical functions be present and have PREDs (Bresnan 2001: 63).

5.1.1 Summary

So far we have been considering simple relative clauses introduced by *aN* and containing a resumptive pronoun, as in example (104). This can be schematized as in (120). An illformed variant, with no resumptive for *aN* to consume, can be schematized as in (122).

(104) an ghirseach a-r ghoid na síogaí í
the girl aN-PAST stole the fairies her
the girl that the fairies stole away

(120) $[_{\text{CP}}$ aN . . . pro]

(122) $*[_{\text{CP}}$ aN . . . t]

We have seen that the lexical entry for *aN* in (121) and the CP and C$'$ rules in (110) and (111) guarantee that (120) is wellformed and that (122) is illformed. So far the analysis requires a relative clause introduced by *aN* to contain a pronoun. The resumptive nature of this pronoun is guaranteed by *aN*'s manager resource, which can only be satisfied if there is a pronominal resource corresponding to the TOPIC of *aN*'s clause. Furthermore, the resource-sensitivity of linear logic guarantees that the glue derivation succeeds if and only if the pronoun is a resumptive, i.e. the pronoun must not be a resource on which there is a dependency (other than *aN*'s).

5.2 *Go* and *aL*

We can now turn our attention to the other two complementizers, *go* and *aL*. The first of these is the complementizer that occurs when there is no long-distance dependency passing through the clause and no resumptive pronoun:

(123) Creidim go-r inis sé bréag.
believe.1SG go-PAST tell he lie
I believe that he told a lie.
(McCloskey to appear: 2, (3))

In its general uninflected form, *go* simply identifies its f-structure with that of its mother, passing up f-structural information through the verbal head path.[23]

(124) *go* C $\uparrow = \downarrow$

By the definitions of grammatical activity in (107), the base form of *go* is neither syntactically nor semantically active. It follows from the Extended Coherence Condition that *go* cannot introduce a relative clause. The ECC requires that the TOPIC in the relative clause be related to some subcategorized grammatical function; but the lexical entry for *go* in

[23]The inflected form of *go* in (123), *gor*, additionally specifies that $(\uparrow \text{ TENSE}) = \text{PAST}$.

(124), unlike the entry for *aN* in (121) above and the entry for *aL* in (127) below, does not link the TOPIC to anything. Therefore, a relative clause must be introduced by *aN* or *aL*, and cannot be introduced by *go*.

The complementizer *aL* indicates the presence of a long distance dependency passing through its clause. It either binds the foot of the long distance dependency, so long as the foot is not a pronoun, or it passes the dependency through its clause:

(125) an fhilíocht a chum sí __
the poetry aL composed she
the poetry that she composed
(McCloskey to appear: 4, (6))

(126) cuid den fhilíocht a chualaís ag do sheanmháthair á rá a cheap an sagart úd __
some of.the poetry aL heard [S2] by your grandmother being-said aL composed the priest DEMON
some of the poetry that you heard your grandmother saying that the priest composed
(McCloskey to appear: 3, (5b))

We get the following three patterns:

(127) $[_{\text{CP}}$ aL ... t]

(128) $[_{\text{CP}}$ aL ... $[_{\text{CP}}$ aL ... t]

(129) $*[_{\text{CP}}$ aL ... pro]

The lexical entry for *aL*, repeated in (130), together with the CP and C′ rules, and other general properties of our theory account for these facts.

$$(130)\quad aL \quad \text{C} \quad \left\{ \begin{array}{l|l} (\uparrow \text{TOPIC}) = (\uparrow \text{COMP TOPIC}) & \begin{array}{l} (\uparrow \text{TOPIC}) = (\uparrow \text{GF}) \\ (\uparrow \text{TOPIC PRED}) = \text{`pro'} \\ (\uparrow \text{TOPIC PRONTYPE}) = \text{variable} \end{array} \end{array} \right\}$$

This lexical entry is disjunctive. With respect to *aL*'s f-structure, the first disjunct identifies the TOPIC and COMP TOPIC. In order for this equation to be satisfied, there must be an f-structure embedded as the value of COMP in *aL*'s f-structure and this embedded f-structure must have a TOPIC. This is a functional control equation, similar to that found in equi and raising, which identifies some outer GF, in this case TOPIC, with some embedded GF. The second disjunct identifies the TOPIC with some GF in *aL*'s f-structure. These two disjuncts have the effect of local TOPIC passing: the bottom-most *aL* identifies the TOPIC with a GF and then any higher *aL*s keep passing up the TOPIC one nuclear f-structure at a time, until the threading terminates at the top of the relative clause. Each TOPIC is identified with a lower TOPIC, until the recursion bottoms out in the lowest TOPIC's f-structure, where the lowest TOPIC is identified with some GF. The ECC is satisfied, because in the lowest f-structure the TOPIC is identified with a subcategorized grammatical function, and all higher TOPICs are too, as they are all identified with the immediately lower TOPIC. This accounts for sentences (125) and (126), which are respectively examples of patterns (127) and (128).

What about pattern (129)? What blocks *aL* from binding a TOPIC that is a (resumptive) pronoun? This is where the second and third equations in the right hand disjunct in (130) comes in. The second equation specifies that the TOPIC PRED is 'pro', but the overt pronoun has its own PRED. This allows the relativized GF to satisfy Completeness. However, the value of PRED is a "semantic form" (Kaplan and Bresnan 1982, Dalrymple 2001) and is always unique. Although both the pronoun and the equation in question are specifying that the TOPIC's PRED is 'pro', these two instances of 'pro' are unique and cannot be identified. The third equation specifies that *aL*'s TOPIC must be a variable, with PRONTYPE var(iable). A personal pronoun's PRONTYPE is specified as pers(onal). The two values are incompatible, thus *aL* cannot bind an overt pronominal topic.

5.3 Mixed Chains

In addition to the one clause cases just discussed, and the case of multiple *aL*s, McCloskey (to appear) discusses what he calls "mixed chains",[24] which have posed various degrees of difficulty for previous analyses of Irish complementizers and relativization. Here are the relevant schematic examples:

(131) Pattern 1: complex NP island with island-internal $\overline{\text{A}}$ -movement and island-external resumptive binding
$[_{\text{CP}}$ aN $[_{\text{TP}} \ldots [_{\text{DP}}$ (D) $[_{\text{NP}}$ N $[_{\text{CP}}$ aL $[_{\text{TP}} \ldots$ __ $\ldots]]]]]$
(McCloskey to appear: 15, (27))

(132) Pattern 2: resumptive pronoun in a position inaccessible to movement
$[_{\text{CP}}$ aL $[_{\text{TP}} \ldots [_{\text{CP}}$ aN $[_{\text{TP}} \ldots$ pro $\ldots]]]]$
(McCloskey to appear: 17, (33))

(133) Pattern 3: a resumptive pronoun with multiple occurrences of the resumptive complementizer.
$[_{\text{CP}}$ aN $[_{\text{TP}} \ldots [_{\text{CP}}$ aN $[_{\text{TP}} \ldots$ pro $\ldots]]]]$
(McCloskey to appear: 19, (40))

Interactions of the lexical entries for *aL* and *aN* with the CP and C′ rules in (110) and (111) account for these cases.

5.3.1 Pattern 1

Pattern one occurs when there is relativization out of a complex NP island. The very existence of this pattern is at first surprising, because by definition, an island should block movement. However, as McCloskey (to appear) notes, complex NP islands are unique in having a position *inside* the island which can host movement: the specifier of the NP complement CP. If *aL* is signalling movement to its SpecCP, the lower part of this pattern is just island-internal $\overline{\text{A}}$ -movement. It would be surprising if we could have *subsequent* movement from this position, as this would be movement out of an island. The following slight variation on pattern one, where there is an upper *aL* marking movement out of the complex NP is therefore unexpected and is in fact unattested:

(134) * $[_{\text{CP}}$ aL $[_{\text{TP}} \ldots [_{\text{DP}}$ (D) $[_{\text{NP}}$ N $[_{\text{CP}}$ aL $[_{\text{TP}} \ldots$ t $\ldots]]]]]$

However, a defining characteristic of resumptive binding is that it *is not* sensitive to islands (McCloskey 1990). So it is not surprising from the perspective of islands that pattern one exists.

There is nevertheless one surprising property of pattern one, on either McCloskey's (to appear) movement approach or the present monostratal LFG account. Namely, if *aN* is signalling the presence of resumption, where is the resumptive pronoun in pattern one? McCloskey (to appear) proposes, within the assumptions of the Minimalist Program (MP; Chomsky 1995), that *aN* and *aL* are actually signalling whether SpecCP is filled by Merge (*aN*) or Move (*aL*) of an operator.

The LFG account that I propose naturally does not countenance the operations of Move versus Merge. Rather, pattern one is explained by the interplay of the lexical entries for *aN* and *aL*, the CP and C′ phrase structure rules, and the theory of the syntax-semantics interface, glue semantics.

Let us consider the following pattern one example:

(135) rud a raibh coinne aige a choimhlíonfadh _ an aimsir
thing aN was expectation at.him aL fulfill.COND the time
something that he expected time would confirm
(McCloskey to appear: 15, (28))

[24] Pattern three qualifies as a mixed chain under McCloskey's (to appear) transformational assumptions, but in the base generated approach I am working with, this classification may seem strange, as there is two *aN*s and nothing seems mixed. However, I have chosen to stick to McCloskey's terminology.

This can be schematically represented as

(136) $[_{\text{CP}}$ aN $[_{\text{TP}} \ldots [_{\text{DP}}$ (D) $[_{\text{NP}}$ N $[_{\text{CP}}$ aL $[_{\text{TP}} \ldots \text{t} \ldots]]]]]$
(McCloskey to appear: 15, (27))

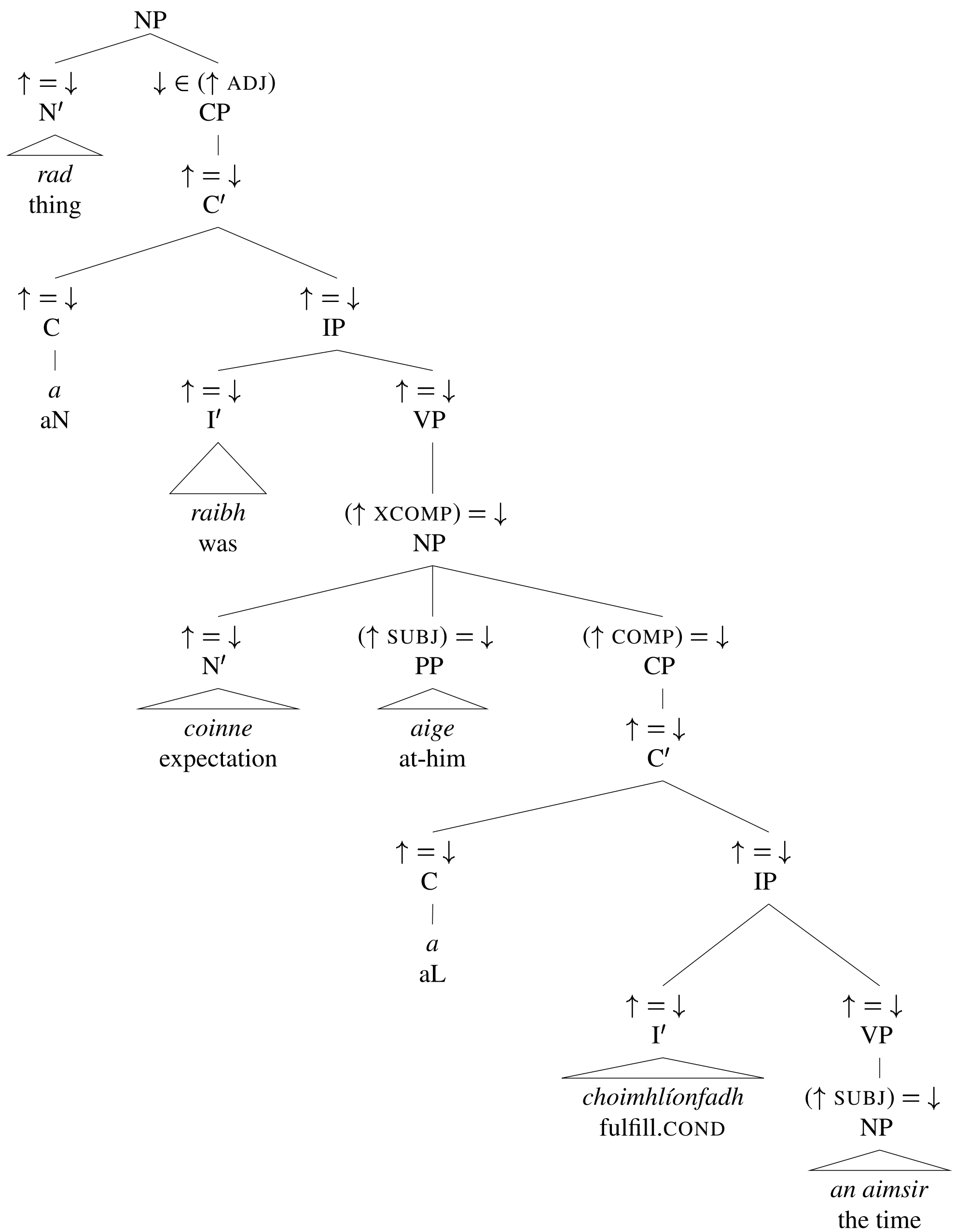
Pattern one relies on the expansion of the C′ rule that realizes the optional equation, as shown in (137). Also, the c-structure for (135) must be constructed by realizing the optional material on the CP rule for the relative CP, but not for the CP complement of *coinne* ('expectation'), as will be explained shortly. This sets up the ANTECEDENT for the TOPIC and provides the meaning constructor abbreviated as **[rel]** for the relative clause.

(110)
$$\text{CP} \longrightarrow \begin{array}{c} \text{C}' \\ \uparrow = \downarrow \\ \left(\begin{array}{c} \textbf{[rel]} \\ ((\uparrow \text{TOPIC})_\sigma\ \text{ANTECEDENT}) = (\text{ADJ}\ \in \uparrow)_\sigma \end{array} \right) \end{array}$$

(137)
$$\text{C}' \longrightarrow \begin{array}{c} \text{C} \\ \uparrow = \downarrow \\ [(\uparrow \text{TOPIC})_\sigma\ \text{ANTECEDENT}] \multimap [(\uparrow \text{TOPIC})_\sigma \otimes ((\uparrow \text{TOPIC})_\sigma\ \text{ANTECEDENT})] \end{array} \quad \begin{array}{c} \text{IP} \\ \uparrow = \downarrow \\ \end{array}$$

Let us assume the following c-structure for (135):

(138)



From this c-structure, assuming we have normal lexical entries for the terminals (the entries for *aN* and *aL* will be discussed in more detail shortly), we get the following f-structure:

(139)
$$f\left[\begin{array}{ll}
\textsc{pred} & \text{‘thing’} \\
\textsc{num} & \text{sg} \\
\textsc{adj} & \left\{ g\left[\begin{array}{ll}
\textsc{pred} & \text{‘be’} \\
\textsc{subj} & h\left[\begin{array}{ll}
\textsc{pred} & \text{‘pro’} \\
\textsc{pers} & 3 \\
\textsc{num} & \text{sg} \\
\textsc{gend} & \text{masc} \\
\textsc{prontype} & \text{pers}
\end{array}\right] \\
\textsc{xcomp} & i\left[\begin{array}{ll}
\textsc{pred} & \text{‘expectation’} \\
\textsc{subj} & \\
\textsc{comp} & j\left[\begin{array}{ll}
\textsc{pred} & \text{‘fulfill’} \\
\textsc{subj} & k\left[\begin{array}{ll}
\textsc{pred} & \text{‘time’} \\
\textsc{spec} & \left[\begin{array}{ll}\textsc{pred} & \text{‘the’}\end{array}\right]
\end{array}\right] \\
\textsc{obj} & l\left[\begin{array}{ll}
\textsc{pred} & \text{‘pro’} \\
\textsc{prontype} & \text{var}
\end{array}\right] \\
\textsc{topic} &
\end{array}\right]
\end{array}\right] \\
\textsc{topic} &
\end{array}\right] \right\}
\end{array}\right]$$

Notice that I am treating *raibh* (‘be’) as a subject raising verb, so that *raibh* identifies its SUBJECT with that of its XCOMP; this is not crucial to the matter at hand. I am also assuming for simplicity that this SUBJECT is a personal pronoun, leaving aside the fact that *aige* (‘at-him’) is a preposition with pronominal inflection.

The lexical information provided by the entries for *aN* and *aL* are repeated in (140) and (141).[25] Note in particular that it is the second disjunct of *aL* that is being realized, as it this *aL* is in the clause containing the foot of the long distance dependency.

(140) *aN* C

$$\begin{aligned}
&(g\ \textsc{topic}) = (g\ [\textsc{gf}^{+} - \textsc{subj}]) \\
&(g\ \textsc{topic}) = (g\ \textsc{xcomp comp obj}) \\
&(g\ \textsc{topic}) = l \\[1em]
&(g\ \textsc{topic prontype}) \\[1em]
&\lambda x \lambda y.x :\quad ((\uparrow \textsc{topic})_\sigma\ \textsc{antecedent}) \multimap ((\uparrow \textsc{topic})_\sigma \multimap ((\uparrow \textsc{topic})_\sigma\ \textsc{antecedent})) \\
&\qquad\qquad = f_\sigma \multimap (l_\sigma \multimap f_\sigma)
\end{aligned}$$

(141) *aL* C

$$\begin{aligned}
&(j\ \textsc{topic}) = (j\ \textsc{gf}) \\
&(j\ \textsc{topic}) = (j\ \textsc{obj}) \\
&(j\ \textsc{topic}) = l \\[1em]
&(j\ \textsc{topic pred}) = \text{‘pro’} \\[1em]
&(j\ \textsc{topic prontype}) = \text{var}
\end{aligned}$$

[25] I have also instantiated the functional equations with nodes from the above f-structure and provided the full version of *aN*'s meaning constructor. Where appropriate, I have spelled out how functional equations are solved in successive steps.

We can see that both TOPICs are identified with the OBJECT of *choimhlíonfadh* ('fulfill'), which is the grammatical function that is relativized. The C′ rule provides a pronominal resource for the OBJECT, although it is not overtly realized as a pronoun. The upper complementizer, *aN*, consumes this pronominal resource. Thus, its glue resource is discharged, along with the pronominal resource provided by the c-structure rule, and the glue proof terminates successfully. The lower complementizer, *aL*, not only identifies its TOPIC with the OBJECT of its clause, it also states that it has PRONTYPE var (i.e., the missing OBJECT is a variable) and provides its TOPIC with a PRED, satisfying Completeness. *AN* checks that its OBJ (via its TOPIC) has a PRONTYPE and this check is satisfied by PRONTYPE var. This has the effect of ensuring that *aN* is only licensed when there is either a resumptive pronoun with PRONTYPE pers or an *aL* complementizer in *aN*'s TOPIC's clause.[26]

This means that we predict the following patterns, including pattern one, schematized more simply as (144), and (142) and (143) that we have discussed before:

(142) $[_{\text{CP}}$ aN ... pro]

(143) *$[_{\text{CP}}$ aN ... *t*]

(144) $[_{\text{CP}}$ aN ... $[_{\text{CP}}$ aL ... *t*]]

(145) $[_{\text{CP}}$ aN ... $[_{\text{CP}}$ go ... $[_{\text{CP}}$ aL ... *t*]]]

(146) *$[_{\text{CP}}$ aN ... $[_{\text{CP}}$ aL ... $[_{\text{CP}}$ go ... *t*]]]

The last two examples are especially interesting. Pattern (145) is unattested, as far as I am aware, but it is predicted by the theory. The reason is that *aN* does not check locally for its TOPIC. Since this complementizer requires a pronominal topic, its requirement is unbounded, as pronominalization in general is unbounded. Therefore, an intervening *go*, which does not pass the TOPIC up, is of no consequence, as *aN* can simply search past it.

However, the situation for *aL* is different, since it is subject to the locality restrictions on long distance dependencies terminating in gaps (i.e., movement in transformational theories). This manifests itself in *aL*'s local TOPIC-passing via the functional control equation in its first disjunct and its second disjunct, which grounds the TOPIC to some GF at the foot of the long distance dependency. Since *go* does no TOPIC passing, an *aL* above a *go* can only realize its second disjunct, as the first disjunct's functional equation could not be satisfied. This means that *aL* would "spuriously" ground its TOPIC to some local GF, rather than to the actual foot of the dependency, which it cannot reach. This would fail for various reasons. First, *aL* would give the spurious GF a PRONTYPE var. If this GF has a conflicting PRONTYPE (e.g., if it's a personal pronoun), there will be a uniqueness violation. Should the GF not have an existing PRONTYPE feature, *aL* will give it one. We could assume that there would be general restrictions against full noun phrases having a PRONTYPE feature.[27] More importantly, though, the second disjunct of *aL* specifies that its TOPIC's PRED is 'pro'. As the TOPIC is identified with the spurious GF, this means that the GF's PRED is also 'pro'. However, in order to independently satisfy Completeness, the GF must already have its own PRED. Even if this PRED were also 'pro' (i.e., if the GF were a pronominal), two PRED values cannot be identified; therefore (146) would be out due to a uniqueness violation

5.3.2 Pattern 2

Pattern two is, as McCloskey (to appear: 18) puts it, the mirror image of pattern one, because here we have a higher *aL* and a lower *aN*:

(147) $[_{\text{CP}}$ aL $[_{\text{TP}}$ ... $[_{\text{CP}}$ aN $[_{\text{TP}}$ ... pro ...]]]]
(McCloskey to appear: 17, (33))

[26] Furthermore, the *aL* in question must be the *aL* that terminates the long distance dependency, as it must provide a PRONTYPE var to the TOPIC and ensure that it has a PRED.

[27] Epithets would be an exception, but they would not have PRONTYPE var in any case. Either they would have PRONTYPE pers, or their own kind of PRONTYPE, depending on what is motivated by the facts.

Sentence (148) is an example of this pattern, which has a pronoun in “a position inaccessible to movement” (McCloskey to appear: 17).

(148) aon duine a cheap sé a raibh ruainne tobac aige
any person aL thought he aN was scrap tobacco at-him
anyone that he thought had a scrap of tobacco
(McCloskey to appear: 17, (34))

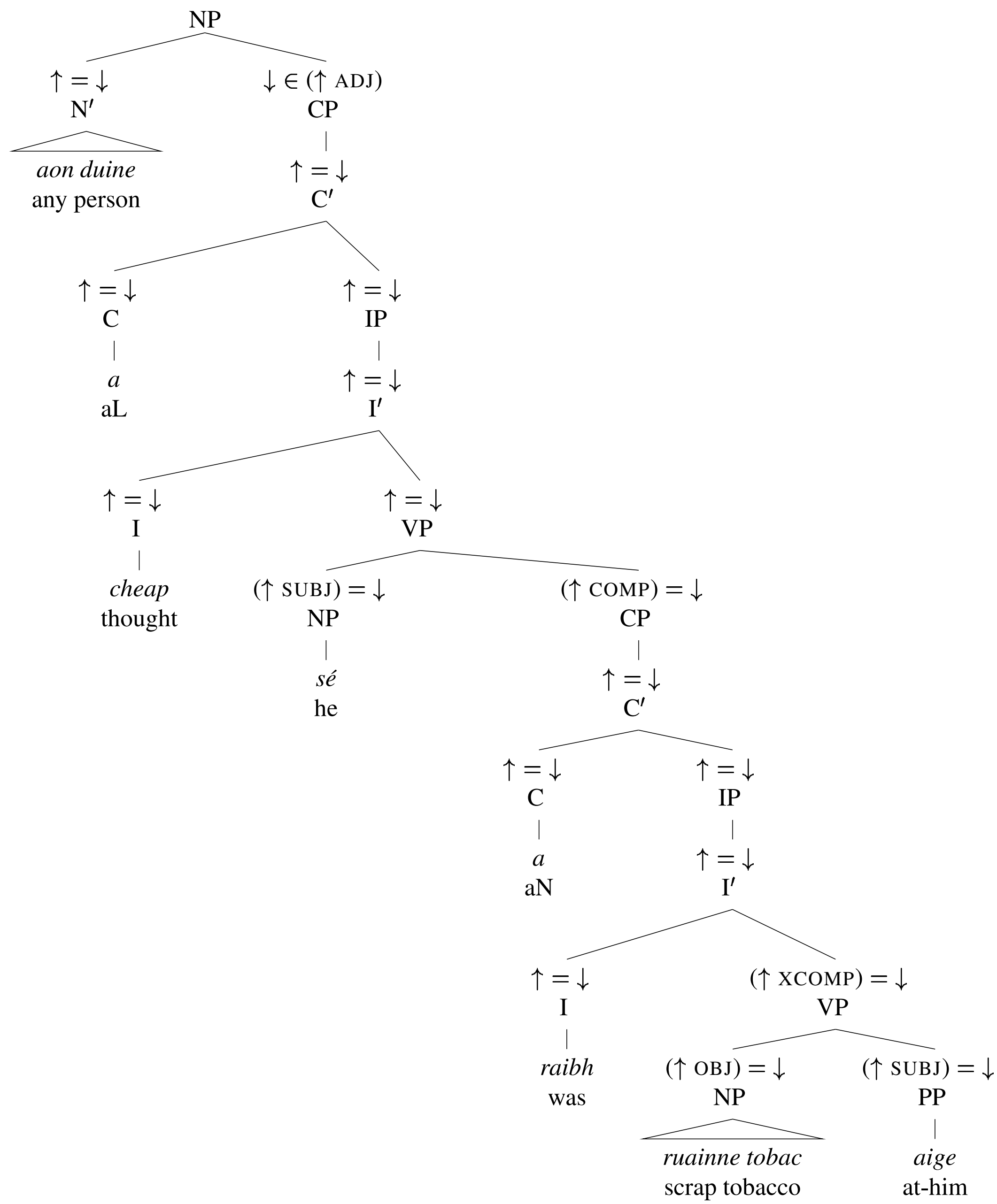
Pattern two requires the C′ rule without its optional meaning constructor; as in pattern one, the CP rule must be realized with the **[rel]** meaning constructor to construct the relative clause, but must be realized without **[rel]** to construct the complement CP of *cheap* (‘thought’):

(149) $$\mathrm{C}' \longrightarrow \begin{array}{cc} \mathrm{C} & \mathrm{IP} \\ \uparrow = \downarrow & \uparrow = \downarrow \end{array}$$

(110) $$\mathrm{CP} \longrightarrow \begin{array}{c} \mathrm{C}' \\ \uparrow = \downarrow \\ \left(\begin{array}{c} \textbf{[rel]} \\ ((\uparrow \textsc{topic})_\sigma\ \textsc{antecedent}) = (\textsc{adj}\ \in \uparrow)_\sigma \end{array} \right) \end{array}$$

I will assume the following c-structure for (148):

(150)



From this c-structure we get the following f-structure:

(151)

$$
f\left[\begin{array}{ll}
\text{PRED} & \text{‘person’}\\
\text{SPEC} & \left[\begin{array}{ll}\text{PRED} & \text{‘any’}\end{array}\right]\\
\text{ADJ} & \left\{ g\left[\begin{array}{ll}
\text{PRED} & \text{‘thought’}\\
\text{SUBJ} & h\left[\begin{array}{ll} \text{PRED} & \text{‘pro’}\\ \text{PERS} & 3\\ \text{NUM} & \text{sg}\\ \text{GEND} & \text{masc}\\ \text{PRONTYPE} & \text{pers}\end{array}\right]\\
\text{COMP} & i\left[\begin{array}{ll}
\text{PRED} & \text{‘be’}\\
\text{SUBJ} & \\
\text{XCOMP} & j\left[\begin{array}{ll}
\text{PRED} & \text{‘have’}\\
\text{SUBJ} & k\left[\begin{array}{ll} \text{PRED} & \text{‘pro’}\\ \text{PERS} & 3\\ \text{NUM} & \text{sg}\\ \text{GEND} & \text{masc}\\ \text{PRONTYPE} & \text{pers}\end{array}\right]\\
\text{OBJ} & l\left[\begin{array}{ll} \text{PRED} & \text{‘tobacco’}\\ \text{ADJ} & \left\{ m\left[\begin{array}{ll}\text{PRED} & \text{‘scrap’}\end{array}\right]\right\}\end{array}\right]
\end{array}\right]\\
\text{TOPIC} &
\end{array}\right]\\
\text{TOPIC} &
\end{array}\right]\right\}
\end{array}\right]
$$

The outer f-structure corresponds to *aon duine* (‘anyone’), the head of the relative clause. The complementizer *aL* introduces the relative clause. The relative clause is headed by the verb *cheap* (‘think’), with a pronominal subject *sé* (‘he’) and a complement CP, whose complementizer is *aN*. As before, *raibh* (‘was’) is a raising verb. Again I have analyzed the inflected preposition simply as a pronoun. To get the innermost f-structure to have *pred* ‘have’, I am assuming either a constructional or configurational (e.g., small clause) analysis, whereby a PP and an NP can form a predicative environment. Alternatively, perhaps the preposition *aig-* (‘at’) should have two arguments, the inflected pronominal SUBJECT and the OBJECT *ruainne tobac* (‘scrap tobacco’); the preposition could then be contributing the PRED ‘have’. I leave these issues aside here and simply assume that the f-structure can be derived, since what really interests us is just the resumptive pronoun.

The lexical entries for the complementizers *aL* and *aN* are repeated in (152) and (153), with the functional equations instantiated with node labels from f-structure (151). Note that only the left disjunct of the general entry for *aL* is realized, as the complementizer is not at the foot of the long distance dependency.

(152) *aL* C $(g\ \text{TOPIC}) = (g\ \text{COMP TOPIC})$
$(g\ \text{TOPIC}) = (i\ \text{TOPIC})$
$(g\ \text{TOPIC}) = k$

(153) *aN* C $(i\ \textsc{topic}) = (i\ [\textsc{gf}^{+} - \textsc{subj}])$
$(i\ \textsc{topic}) = (i\ \textsc{xcomp subj}])$
$(i\ \textsc{topic}) = k$

$(i\ \textsc{topic prontype})$

$$\lambda x \lambda y.x : \quad ((\uparrow \textsc{topic})_{\sigma}\ \textsc{antecedent}) \multimap ((\uparrow \textsc{topic})_{\sigma} \multimap ((\uparrow \textsc{topic})_{\sigma}\ \textsc{antecedent})) = f_{\sigma} \multimap (k_{\sigma} \multimap f_{\sigma})$$

Again we see that the clauses introduced by *aL* and *aN* share the same TOPIC. *AN* identifies the TOPIC initially, in its normal unbounded manner (although it does not have to look very far in this case): it is the resumptive subject of the innermost clause. *AN* checks that this is a pronominal, with a PRONTYPE, and this check succeeds as personal pronouns are lexically specified with this feature. Lastly, *aN* consumes the pronominal resource introduced by this pronoun, satisfying its own manager resource and allowing the rest of the proof to proceed as if the pronoun were not there. *AL* merely needs to identify its clause's TOPIC with the TOPIC of its COMP.

Notice that the right disjunct of *aL* cannot be realized here, for the same reason that barred pattern (146), which is repeated below.

(146) $*[_{\text{CP}}$ aN … $[_{\text{CP}}$ aL … $[_{\text{CP}}$ go … *t*]]]

There is no relativized grammatical function in *aL*'s clause (i.e., there is no trace in transformational terms). The right disjunct of *aL* identifies its TOPIC with some local GF. But, since the right disjunct has the equation $(\uparrow\ \textsc{topic pred}) =$ 'pro', there will be a uniqueness violation, as this value is a semantic form and cannot be identified with any other value and all the other local GFs must have their own PRED values in order to satisfy Completeness (Kaplan and Bresnan 1982, Bresnan 2001, Dalrymple 2001).

The result is the following predictions:

(154) $[_{\text{CP}}$ aL … $[_{\text{CP}}$ aN … pro]]

(155) $*[_{\text{CP}}$ aL … $[_{\text{CP}}$ go … $[_{\text{CP}}$ aN … pro]]]

(154) is pattern two, the subject of this section. The second example has a *go* intervening between *aL* and *aN*. This is predicted to be ungrammatical, essentially for the same reasons as (146). *Go* does no TOPIC passing and therefore the left disjunct of *aL* cannot be satisfied. But neither can the right disjunct, because it is seeking to assign a PRED with the value of 'pro' to whatever local GF it has identified with its TOPIC. This GF must independently have a PRED already, to satisfy Completeness, and since PRED values are unique and cannot be identified, the equation in question cannot be satisfied.

5.3.3 Pattern 3

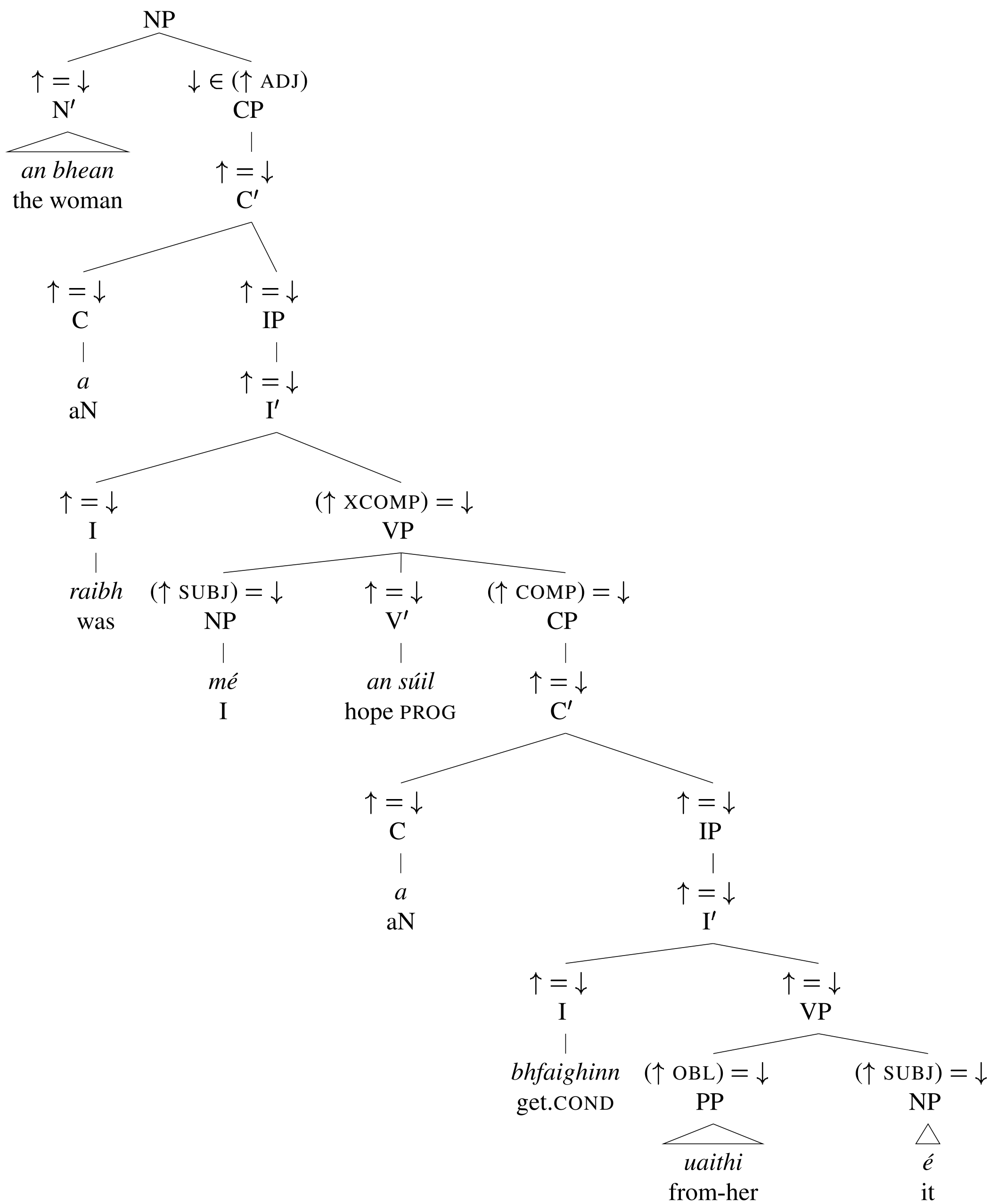
The third pattern that McCloskey (to appear) discusses involves a resumptive pronoun, again in a position inaccessible to movement, with a local *aN* as would be expected, but also with a higher *aN*, as in pattern one:

(156) an bhean a raibh mé ag súil a bhfaighinn uaithi é
the woman aN was I hope PROG aN get.COND.S1 from-her it
the woman that I was hoping that I would get it from (her)
(McCloskey to appear: 19, (41))

From the resource management perspective, this pattern is like pattern one, in seemingly not having enough pronominal resources to go around. Unsurprisingly, the solution is the same as in pattern one: the C′ rule must realize its optional pronominal meaning constructor. The CP rule will operate as usual, contributing **[rel]** for the relative clause, but not for the complement clause.

Let us assume the following c-structure for (156):

(157)



This yields the f-structure in (158).

(158)

$$
f\left[\begin{array}{ll}
\textsc{pred} & \text{`woman'} \\
\textsc{spec} & \left[\begin{array}{ll}\textsc{pred} & \text{`the'}\end{array}\right] \\
\textsc{adj} & \left\{ g\left[\begin{array}{ll}
\textsc{pred} & \text{`be'} \\
\textsc{subj} & \\
\textsc{xcomp} & h\left[\begin{array}{ll}
\textsc{pred} & \text{`hope'} \\
\textsc{subj} & i\left[\begin{array}{ll}
\textsc{pred} & \text{`pro'} \\
\textsc{pers} & 1 \\
\textsc{num} & \text{sg} \\
\textsc{prontype} & \text{pers}
\end{array}\right] \\
\textsc{comp} & j\left[\begin{array}{ll}
\textsc{pred} & \text{`get'} \\
\textsc{subj} & \\
\textsc{obj} & k\left[\begin{array}{ll}
\textsc{pred} & \text{`pro'} \\
\textsc{pers} & 3 \\
\textsc{num} & \text{sg} \\
\textsc{gend} & \text{neut} \\
\textsc{prontype} & \text{pers}
\end{array}\right] \\
\textsc{obl} & l\left[\begin{array}{ll}
\textsc{pred} & \text{`from'} \\
\textsc{obj} & m\left[\begin{array}{ll}
\textsc{pred} & \text{`pro'} \\
\textsc{pers} & 1 \\
\textsc{num} & \text{sg} \\
\textsc{gend} & \text{fem} \\
\textsc{prontype} & \text{pers}
\end{array}\right]
\end{array}\right] \\
\textsc{topic} &
\end{array}\right]
\end{array}\right] \\
\textsc{topic} &
\end{array}\right] \right\}
\end{array}\right]
$$

Here are the instantiated lexical entries for the two instances of ***aN***:

(159) aN_1 C

$(g\ \textsc{topic}) = (g\ [\textsc{gf}^+ - \textsc{subj}])$
$(g\ \textsc{topic}) = (g\ \textsc{xcomp comp obl obj})$
$(g\ \textsc{topic}) = (h\ \textsc{comp obl obj})$
$(g\ \textsc{topic}) = (j\ \textsc{obl obj})$
$(g\ \textsc{topic}) = (l\ \textsc{obj})$
$(g\ \textsc{topic}) = m$

$(g\ \textsc{topic prontype})$

$$\lambda x \lambda y.x : \quad ((\uparrow \textsc{topic})_\sigma\ \textsc{antecedent}) \multimap ((\uparrow \textsc{topic})_\sigma \multimap ((\uparrow \textsc{topic})_\sigma\ \textsc{antecedent}))$$
$$= f_\sigma \multimap (m_\sigma \multimap f_\sigma)$$

(160) aN_2 C $(j\ \text{TOPIC}) = (j\ [\text{GF}^{+} - \text{SUBJ}])$
$(j\ \text{TOPIC}) = (j\ \text{OBL OBJ}])$
$(j\ \text{TOPIC}) = m$

$(j\ \text{TOPIC PRONTYPE})$

$$\lambda x \lambda y.x : \quad ((\uparrow \text{TOPIC})_\sigma\ \text{ANTECEDENT}) \multimap ((\uparrow \text{TOPIC})_\sigma \multimap ((\uparrow \text{TOPIC})_\sigma\ \text{ANTECEDENT})) = f_\sigma \multimap (m_\sigma \multimap f_\sigma)$$

Both complementizers identify the TOPIC of their clause with the pronominal complement of the preposition in the lowest clause. They both do this in an unbounded manner, as we have already observed. Importantly, both complementizers also contribute a manager resource that is seeking to consume the pronominal resource of the TOPIC. However, since there is only one pronoun, contributing one resource, it seems that one or the other of these manager resources cannot be satisfied.

This is where the annotated C′ rule comes in, with the optional equation realized:

(161)
$$\text{C}' \longrightarrow \underset{[(\uparrow \text{TOPIC})_\sigma\ \text{ANTECEDENT}] \multimap [(\uparrow \text{TOPIC})_\sigma \otimes ((\uparrow \text{TOPIC})_\sigma\ \text{ANTECEDENT})]}{\underset{\uparrow = \downarrow}{\text{C}}} \quad \underset{\uparrow = \downarrow}{\text{IP}}$$

One of the *aN*s consumes the pronominal provided by the actual resumptive pronoun, while the other consumes the pronominal provided by the rule. Notice that one expansion of the rule must provide this resource and the other must not: since there are two *aN*s and one pronominal, only one pronominal resource must be contributed by the rule, otherwise there would be an unused pronominal resource and the resource-sensitivity of linear logic would cause the glue proof to fail.

5.4 Summary

The basic intuition of this account of Irish resumptive pronouns is that there is a one to one correspondence between *aN* and pronominal resources. If there are multiple *aN*s in a sentence, such that there is only one resumptive pronoun, all but one *aN*s must be consuming a pronominal resource provided by the C′ rule. Another intuition is that the *aN*-pronoun relationship is unbounded, since it is a pronominal relationship and as a result is not cyclic, whereas the *aL*-gap relationship is locally bounded and cyclic, as are long distance dependency relations in general. The third intuition is that syntax and semantics place separate requirements on wellformedness and there can be complex interactions between these requirements. For example, although *aN* could introduce a clause with a gap rather than a resumptive, satisfying its manager resource by taking a pronominal resource from the C′ rule, such sentences are ruled out by the syntax, as the gap would not have a PRED or a PRONTYPE as these are provided by the lowermost *aL*. On the flip side, a clause containing a resumptive pronoun must be introduced by *aN* because this is the only way that the rules of semantic compositon (the linear logic side of the glue proofs) can be satisfied, since otherwise there would be an unconsumed pronominal resource.

6 Conclusion

The resource-sensitivity of glue semantics and the resource management approach to resumption in anaphoric control, copy raising, and resumptive pronouns sheds light on resumption as a general phenomenon, rather than keeping it reserved to bound variables of operators. Anaphoric control, copy raising, and resumption all involve resource management of pronouns that are not semantically necessary. Resumption is formally defined as

(162) the use of a manager resource $(A \multimap B) \multimap A$ to consume an anaphoric resource $A \multimap (A \otimes B)$, leaving a modifier on the antecedent $A \multimap A$.

This means that resumption must occur in specific environments, since it is dependent on the presence of manager resources. Manager resources are associated with specific lexical items. The analysis therefore treats typological variation for resumption as a matter of lexical inventories. Diverse linguistic theories have converged on the desirability of locating language variation in the lexicon. The modularity of glue semantics means that the results reported here should be readily applicable in other frameworks.

The dissertation answers several central theoretical questions about resumption. In terms of the syntax, resumption is the normal relationship that holds between an antecedent and an anaphor. In terms of the semantics, resumption is the use of a pronoun or epithet to establish an anaphoric dependency on some antecedent while leaving the argument that the anaphor corresponds to unresolved. Anaphors in particular are used for resumption because of 1) their formal properties, 2) how they recieve their meaning, and 3) their lack of inherent meaning. Languages use resumptions to 1) fulfill syntactic requirements while allowing certain interpretations, or 2) as one possible strategy to establish an anaphoric dependency between a dislocated element and an anaphor occupying the element's position as an argument to some predicate. For a language to display resumption in a given environment, there has to be a manager resource in the environment of the anaphor that removes the anaphor's meaning from consideration for interpretation. This is how differences between languages with respect to resumption are reduced to lexical differences. Lastly, the analysis treats resumptive pronouns like regular pronouns, in the sense that the pronouns themselves simply contribute a normal pronominal resource and pick up an antecedent. The fact that languages do not have special resumptive paradigms follows from this analysis, since resumptive pronouns are not special in any way.

A Proofs with meaning terms (for (117)–(119))

(163) Construction of the relative clause semantics

$$
\dfrac{
\dfrac{
\lambda y.girl(y) : v \multimap r
\qquad
\dfrac{
\dfrac{
\dfrac{
\lambda Q.the(x,fairy(x),Q(x)) : \forall H.[(h_\sigma \multimap H) \multimap H]
\qquad
\dfrac{[u : i_\sigma]^1 \quad \lambda y\lambda x.steal(x,y) : (i_\sigma \multimap (h_\sigma \multimap g_\sigma))}{\lambda x.steal(x,u) : h_\sigma \multimap g_\sigma}\multimap_{\mathcal{E}}
}{the(x,fairy(x),steal(x,u)) : g_\sigma}\multimap_{\mathcal{E}}, H = g
}{\lambda u.the(x,fairy(x),steal(x,u)) : i_\sigma \multimap g_\sigma}\multimap_{\mathcal{I},1}
\qquad
\lambda P\lambda Q\lambda z.P(z) \wedge Q(z) : (i_\sigma \multimap g_\sigma) \multimap [(v \multimap r) \multimap (v \multimap r)]
}{\lambda Q\lambda z.the(x,fairy(x),steal(x,z)) \wedge Q(z) : (v \multimap r) \multimap (v \multimap r)}\multimap_{\mathcal{E}}
}{\lambda z.the(x,fairy(x),steal(x,z)) \wedge girl(z) : v \multimap r}\multimap_{\mathcal{E}}
\qquad
\lambda P\lambda Q.the(y,P(y),Q(y)) : (v \multimap r) \multimap \forall G.[(f_\sigma \multimap G) \multimap G]
}{\lambda Q.the(y,the(x,fairy(x),steal(x,y)) \wedge girl(y),Q(y)) : \forall G.[(f_\sigma \multimap G) \multimap G]}\multimap_{\mathcal{E}}
$$

(164) Pronoun consumption

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{[u : f_\sigma]^2 \quad \lambda x\lambda y.x : f_\sigma \multimap (i_\sigma \multimap f_\sigma)}{\lambda y.u : i_\sigma \multimap f_\sigma}\multimap_{\mathcal{E}}
\quad [w : i_\sigma]^3
}{u : f_\sigma}\multimap_{\mathcal{E}}
\qquad
\dfrac{[z : f_\sigma]^4 \quad \lambda y.y \times y : f_\sigma \multimap (i_\sigma \otimes f_\sigma)}{z \times z : i_\sigma \otimes f_\sigma}\multimap_{\mathcal{E}}
}{let\ z \times z\ be\ u \times w\ in\ u : f_\sigma}\otimes_{\mathcal{E},2,3}
}{z : f_\sigma}
$$

(165) Putting the two parts together

$$
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{[P : f_\sigma \multimap J]^5 \quad z : f_\sigma}{P(z) : J}\multimap_{\mathcal{E}}
}{\lambda z.P(z) : f_\sigma \multimap J}\multimap_{\mathcal{I},4}
\qquad
\lambda Q.the(y,the(x,fairy(x),steal(x,y)) \wedge girl(y),Q(y)) : \forall G.[(f_\sigma \multimap G) \multimap G]
}{the(y,the(x,fairy(x),steal(x,y)) \wedge girl(y),P(y)) : J}\multimap_{\mathcal{E}}, G = J
}{\lambda P.the(y,the(x,fairy(x),steal(x,y)) \wedge girl(y),P(y)) : (f_\sigma \multimap J) \multimap J}\multimap_{\mathcal{I},5}
}{\lambda Q.Q(\lambda P.the(y,the(x,fairy(x),steal(x,y)) \wedge girl(y),P(y))) : \forall K.[(f_\sigma \multimap K)] \multimap K}\forall_{\mathcal{I}}
}{\lambda P.the(y,the(x,fairy(x),steal(x,y)) \wedge girl(y),P(y)) : \forall K.[(f_\sigma \multimap K)] \multimap K}
$$

References

Andrews, Avery (1982). The representation of case in Modern Icelandic. In Bresnan (1982b), (pp. 427–503).

Aoun, Joseph, Lina Choueiri, and Norbert Hornstein (2001). Resumption, movement, and derivational economy. *Linguistic Inquiry*, *32*, 371–403.

Asudeh, Ash (2000). Functional identity and resource-sensitivity in control. In Miriam Butt and Tracy Holloway King (eds.), *Proceedings of the LFG 2000 conference*, Stanford, CA. CSLI Publications. http://csli-publications.stanford.edu/LFG/.

Asudeh, Ash (2002). A resource-sensitive semantics for equi and raising. In David Beaver, Stefan Kaufmann, Brady Clark, and Luis Casillas (eds.), *The construction of meaning*. Stanford, CA: CSLI Publications.

Asudeh, Ash, and Richard Crouch (2001). Glue semantics for HPSG. In Frank van Eynde, Lars Hellan, and Dorothee Beermann (eds.), *Proceedings of the HPSG-2001 conference*, Stanford, CA. CSLI Publications.

Bresnan, Joan (1982a). Control and complementation. *Linguistic Inquiry*, *13*, 343–434.

Bresnan, Joan (ed.) (1982b). *The mental representation of grammatical relations*. Cambridge, MA: MIT Press.

Bresnan, Joan (2001). *Lexical-Functional Syntax*. Oxford: Blackwell.

Bresnan, Joan, and Sam Mchombo (1987). Topic, pronoun, and agreement in chichewâ. *Language*, *63*, 741–782.

Butt, Miriam, Tracy Holloway King, Maria-Eugenia Niño, and Frédérique Segond (1999). *A grammar writer's cookbook*. Stanford, CA: CSLI Publications.

Chierchia, Gennaro (1984a). Anaphoric properties of infinitives and gerunds. In Mark Cobler, Susannah MacKaye, and Michael Wescoat (eds.), *Proceedings of the third West Coast Conference on Formal Linguistics*, (pp. 28–39), Stanford, CA. Stanford Linguistics Association.

Chierchia, Gennaro (1984b). *Topics in the syntax and semantics of infinitives and gerunds*. Ph.D. thesis, University of Massachusetts, Amherst.

Chomsky, Noam (1973). Conditions on transformations. In Stephen Anderson and Paul Kiparsky (eds.), *A festschrift for Morris Halle*, (pp. 233–285). New York: Holt, Rinehart and Winston.

Chomsky, Noam (1993). A minimalist program for linguistic theory. In Kenneth Hale and Samuel Jay Keyser (eds.), *The view from building 20*, (pp. 1–52). Cambridge, MA: MIT Press.

Chomsky, Noam (1995). *The minimalist program*. MIT Press.

Crouch, Richard, and Josef van Genabith (1999). Context change, underspecification, and the structure of glue language derivations. In Dalrymple (1999), (pp. 117–189).

Crouch, Richard, and Josef van Genabith (2000). Linear logic for linguists. Ms., Xerox PARC and Dublin City University.

Dalrymple, Mary (ed.) (1999). *Semantics and syntax in Lexical Functional Grammar: The resource logic approach*. Cambridge, MA: MIT Press.

Dalrymple, Mary (2001). *Lexical Functional Grammar*. San Diego, CA: Academic Press.

Dalrymple, Mary, John Lamping, Fernando Pereira, and Vijay Saraswat (1999a). Overview and introduction. In Dalrymple (1999), (pp. 1–38).

Dalrymple, Mary, John Lamping, Fernando Pereira, and Vijay Saraswat (1999b). Quantification, anaphora, and intensionality. In Dalrymple (1999), (pp. 39–89).

Falk, Yehuda N. (2001). *Lexical-Functional Grammar: An introduction to parallel constraint-based syntax*. Stanford, CA: CSLI Publications.

Fassi-Fehri, Abdelkader (1988). Agreement in Arabic, binding, and coherence. In Michael Barlow and Charles A. Ferguson (eds.), *Agreement in natural language: Approaches, theories, descriptions*. Stanford, CA: CSLI Publications.

Frank, Anette, and Josef van Genabith (2001). LL-based semantics construction for LTAG — and what it teaches us about the relation between LFG and LTAG. In *Proceedings of the LFG 2001 conference*. CSLI Publications.

Girard, Jean-Yves (1987). Linear logic. *Theoretical Computer Science*, *50*, 1–102.

Hashemipour, Margaret Marie (1985). Control theory and *pro* in Modern Persian. Ms., UCSD.

Hashemipour, Margaret Marie (1989). *Pronominalization and control in Modern Persian*. Ph.D. thesis, University of California, San Diego.

Heycock, Caroline (1994). *Layers of predication*. New York: Garland.

Howard, W.A. (1980). The formulae-as-types notion of construction. In J.P. Seldin and J.R. Hindley (eds.), *To H.B. Curry: Essays on combinatory logic, lambda calculus and formalism*, (pp. 479–490). London: Academic press.

Kaplan, Ronald M. (1987). Three seductions of computational psycholinguistics. In P. Whitelock, M. M. Wood, H. L. Somers, R. Johnson, and P. Bennett (eds.), *Linguistic theory and computer applications*, (pp. 149–181). London: Academic Press. Reprinted in Dalrymple et al. (1995), 339–367.

Kaplan, Ronald M. (1995). The formal architecture of Lexical-Functional Grammar. In Mary Dalrymple, Ronald M. Kaplan, John T. Maxwell, and Annie Zaenen (eds.), *Formal issues in Lexical-Functional Grammar*, (pp. 7–27). Stanford, CA: CSLI.

Kaplan, Ronald M., and Joan Bresnan (1982). Lexical-Functional Grammar: A formal system for grammatical representation. In Bresnan (1982b), (pp. 173–281).

Kaplan, Ronald M., and John T. Maxwell (1996). *LFG grammar writer's workbench*. Tech. rep., Xerox PARC, Palo Alto, CA. URL ftp://ftp.parc.xerox.com/pub/lfg/lfgmanual.ps.

Lappin, Shalom (1984). Predication and raising. In Charles Jones and Peter Sells (eds.), *Proceedings of NELS 14*, (pp. 236–252), Amherst, MA. GLSA.

Maling, Joan (1983). Transitive adjectives: A case of categorial reanalysis. In Frank Heny and B. Richards (eds.), *Linguistic categories: Auxiliaries and related puzzles*, (pp. 253–289). Dordrecht: Foris.

McCloskey, James (1979). *Transformational syntax and model theoretic semantics: A case-study in Modern Irish*. Dordrecht: Reidel.

McCloskey, James (1990). Resumptive pronouns, $\overline{\text{A}}$-binding and levels of representation in Irish. In Randall Hendrick (ed.), *Syntax of the Modern Celtic languages*, vol. 23 of *Syntax and Semantics*, (pp. 199–248). San Diego, CA: Academic Press.

McCloskey, James (to appear). Resumption, successive cyclicity, and the locality of operations. In Samuel Epstein and T. Daniel Seeley (eds.), *Prospects for derivational explanation*. Oxford: Blackwell.

McCloskey, James, and Peter Sells (1988). Control and A-chains in Modern Irish. *Natural Language and Linguistic Theory*, *6*, 143–189.

Pollard, Carl, and Ivan A. Sag (1994). *Head-Driven Phrase Structure Grammar*. Chicago, IL and Stanford, CA: The University of Chicago Press and CSLI Publications.

Potsdam, Eric, and Jeffrey T. Runner (2002). Richard returns: Copy Raising and its implications. In Mary Andronis, Chris Ball, Heidi Elston, and Sylvain Neuvel (eds.), *Proceedings of the Chicago Linguistic Society 37*, (p. ??), Chicago, IL. Chicago Linguistic Society.

Rogers, Andy (1971). Three kinds of physical perception verbs. In *Papers from the seventh regional meeting of the Chicago Linguistic Society*, (pp. 206–222).

Rogers, Andy (1972). Another look at flip perception verbs. In *Papers from the eighth regional meeting of the Chicago Linguistic Society*, (pp. 303–315).

Rogers, Andy (1974a). *Physical perception verbs in English: A study in lexical relatedness*. Ph.D. thesis, UCLA.

Rogers, Andy (1974b). A transderivational constraint on Richard? In *Papers from the tenth regional meeting of the Chicago Linguistic Society*, (pp. 551–558).

Sag, Ivan A., and Carl Pollard (1991). An integrated theory of complement control. *Language*, *67*, 63–113.

Sells, Peter (1984). *Syntax and semantics of resumptive pronouns*. Ph.D. thesis, University of Massachusetts, Amherst.

Ura, Hiroyuki (1998). Checking, economy, and copy-raising in igbo. *Linguistic Analysis*, *28*, 67–88.

van Genabith, Josef, and Richard Crouch (1999). How to glue a donkey to an f-structure. In H Bunt and Reinhard Muskens (eds.), *Computing meaning*, vol. 1. Dordrecht: Kluwer.

Zaenen, Annie (1980). *Extraction rules in Icelandic*. Ph.D. thesis, Harvard University. Reprinted as Zaenen (1985).

Zaenen, Annie (1985). *Extraction rules in Icelandic*. New York, NY: Garland Publishing.

Zec, Draga (1987). On obligatory control in clausal complements. In Masayo Iida, Stephen Wechsler, and Draga Zec (eds.), *Working papers in grammatical theory and discourse structure, vol. I: Interactions of morphology, syntax, and discourse*, (pp. 139–168). Stanford, CA: CSLI Publications.