Linguistics and Philosophy (2005) 28:375–446
DOI 10.1007/s10988-005-2656-7 © Springer 2005

ASH ASUDEH

RELATIONAL NOUNS, PRONOUNS, AND RESUMPTION★

ABSTRACT. This paper presents a variable-free analysis of relational nouns in Glue Semantics, within a Lexical Functional Grammar (LFG) architecture. Relational nouns and resumptive pronouns are bound using the usual binding mechanisms of LFG. Special attention is paid to the bound readings of relational nouns, how these interact with genitives and obliques, and their behaviour with respect to scope, crossover and reconstruction. I consider a puzzle that arises regarding relational nouns and resumptive pronouns, given that relational nouns can have bound readings and resumptive pronouns are just a specific instance of bound pronouns. The puzzle is: why is it impossible for bound implicit arguments of relational nouns to be resumptive? The puzzle is highlighted by a well-known variety of variable-free semantics, where pronouns and relational noun phrases are identical both in category and (base) type. I show that the puzzle also arises for an established variable-based theory. I present an analysis of resumptive pronouns that crucially treats resumptives in terms of the resource logic *linear logic* that underlies Glue Semantics: a resumptive pronoun is a perfectly ordinary pronoun that constitutes a surplus resource; this surplus resource requires the presence of a resumptive-licensing resource consumer, a *manager resource*. Manager resources properly distinguish between resumptive pronouns and bound relational nouns based on differences between them at the level of semantic structure. The resumptive puzzle is thus solved. The paper closes by considering the solution in light of the hypothesis of direct compositionality. It is argued that a directly compositional version of the theory is possible, although perhaps not desirable. The implications for direct compositionality are considered.

1. Introduction

Relational nouns, such as *neighbour*, *mother*, and *rumour*, present a challenge to syntax and semantics because, unlike epithets, they behave like ordinary common nouns in terms of both their external

★ For their helpful comments and advice on this paper, I would like to thank David Beaver, Dick Crouch, Mary Dalrymple, Chris Kennedy, Angelika Kratzer, Jim McCloskey, Valeria de Paiva, Ivan Sag, Peter Sells, Ida Toivonen, the audience at LSA 2004, and especially Chris Potts. Daniel Büring and two anonymous *L&P* referees provided invaluable criticism and feedback on previous drafts. All remaining errors are my own.

and internal syntax, yet give rise to ‘bound variable’ readings, as in (1), even though such readings are normally associated only with pronouns and epithets.

(1) Every suburbanite knows a neighbour.

One might expect the natural reading of this sentence to be that every suburbanite knows some neighbour or other, which is a possible reading of the sentence if the relational noun *neighbour* is replaced by a common noun such as *real estate agent*. However, this reading is completely unavailable: the neighbour in question must at least be the neighbour *of* somebody. In fact, sentence (1) does not even readily have the reading on which every suburbanite knows a neighbour of somebody or other. Rather, the natural reading is one where every suburbanite knows one of his/her *own* neighbours. This is what one might pretheoretically call the *bound reading* of the relational noun *neighbour*.

Nouns like *neighbour* are inherently relational, in the sense that the relation is part of their lexical meaning (Barker 1991, 1995, Barker and Dowty 1993). There are also relational modifiers, such as *local* (Mitchell 1986; Partee 1989), which give rise to bound readings:

(2) Every man frequents a local bar.

The natural reading of this sentence is one where each man frequents a bar that is local to him.

Lastly, there are non-relational nouns, such as *mantel*, which normally behave like other common nouns in lacking bound readings, as in (3), but can gain such a reading contextually, as in (4) (Barker 1995, p. 53, fn. 6; Jacobson 1999, p. 145, (43a)):

(3) Everyone noticed the mantel.

(4) Everyone in Berkeley puts eucalyptus leaves on the mantel.

Sentence (3) does not mean that everyone noticed their own mantel, but rather some mantel in the discourse. In contrast, (4) on its most natural reading does mean that everyone in Berkeley puts eucalyptus leaves on their own mantel. The genericity of the present tense is no doubt a factor, as a bound reading is more readily available for (3) if *noticed* is replaced with *notices*. Let us call nominals that have a relational meaning *relational noun phrases*, whether the relation is contributed by the head noun, a modifier, or arises contextually.

Lexically relational nouns like *neighbour* will here exemplify the general class.[1]

The bound readings displayed by relational nouns are normally the reserve of pronominals, as in (5). Noun phrases headed by non-relational nouns normally require a pronominal specifier to give rise to a similar bound reading (except in the case of contextually relational readings), as shown in (6):

(5) Every senator claimed that Bill insulted him.

(6) Every suburbanite gets annoyed with his real estate agent.

From a slightly different perspective, then, the question to ask about relational noun phrases is why there is no overt pronoun required for their bound reading. The answer lies in the key difference between relational nouns and non-relational nouns: the former have an implicit argument, the latter do not. It is the implicit argument that gets bound, not the noun phrase as a whole (Mitchell 1986; Partee 1989).

Given the similarities between bound readings of relational nouns and pronouns, it seems pretheoretically desirable to account for all bound readings using the same mechanism. This paper presents a new analysis of relational nouns that indeed accounts for their bound reading using the mechanism for pronominal binding. It further shows that the analysis of relational nouns fits well with a semantics of genitives and relational arguments realized in PP complements. The analysis also makes correct predictions with respect to quantifier scope, crossover, and reconstruction for bound relational nouns.

However, the binding mechanism that applies to relational nouns and pronouns cannot be completely undiscriminating. In particular, it cannot allow bound relational noun phrases to occur wherever bound pronouns do. A specific instance of this problem arises in the case of resumptive pronouns. Using English words

[1] Note that I use the term *noun phrase* and the category NP as labels for full nominal expressions, not just for complements of a category D (determiner) that heads a determiner phrase (DP; Abney 1987). The NP/DP distinction is not relevant to most of what I have to say and will only be discussed explicitly where it is. Readers who feel more comfortable reading NP as DP should do so.

solely for expository purposes, the following examples highlight the problem:

(7) Every suburbanite who Mary knows that he got arrested vanished.

(8) *Every suburbanite who Mary knows that a neighbour got arrested vanished.

The impossible intended meaning of (8) is that every suburbanite who is such that Mary knows that a neighbour of the suburbanite got arrested vanished.

The resulting puzzle is this:

(9) If the implicit argument of a relational noun can be bound like a pronoun, why is it impossible for the argument to function resumptively?

Let us call this the *resumptive puzzle* about relational nouns. The problem is first mentioned briefly by Jacobson (1999, p. 130, fn. 10), in the context of a variable-free treatment of pronouns, but I show that it also occurs in the variable-based account of Heim and Kratzer (1998).

Many languages, e.g., Arabic, Irish, Hebrew, and Swedish (Engdahl 1985; McCloskey 1990, 2002; Shlonsky 1992), allow the equivalent of (7), but none of these have been reported as allowing the equivalent of (8). This is shown by the following contrast in similar Swedish sentences:

(10)

Varje förortsbo	som	Maria	vet	att	han
every suburbanite	that	M.	knows	that	he
arresterades	försvann.				
arrest.PASSIVE	vanished				

Every suburbanite who Maria knows that he was arrested vanished.

(11)

*Varje förortsbo	som	Maria	vet	att	en	granne
every suburbanite	that	M.	knows	that	a	neighbour
arresterades	försvann.					
arrest.PASSIVE	vanished					

Every suburbanite who Maria knows that a neighbour was arrested vanished.

The equivalent contrast also occurs in questions:

(12) Vilken förortsbo vet Maria att han försvann?
which suburbanite knows M. that he vanished
Which suburbanite does Maria know that he vanished?

(13) *Vilken förortsbo vet Maria att en granne
which suburbanite knows M. that a neighbour
försvann?
vanished
Which suburbanite does Maria know that a neighbour vanished?

Engdahl (1985) and Asudeh (2004) argue that subject pronouns like those in (10) and (12) are true resumptive pronouns in Swedish. However, a sentence with a relational noun phrase occupying the equivalent position is completely ungrammatical and is difficult to interpret even purely pragmatically.

I adopt the theory of resumptive pronouns presented in Asudeh (2004) and show that the analyses of relational nouns and resumptives together solve the resumptive puzzle. The key to the solution is that implicit arguments to relational nouns are semantic arguments but not syntactic arguments. Implicit relational arguments are present only at a level of semantic representation and are not present in the syntax. In contrast, resumptive pronouns are syntactic arguments, like pronouns more generally. Resumptive licensers can only license syntactic arguments, which means that implicit relational arguments cannot be licensed as resumptives. The analysis thus formally captures the intuition of Partee (1989) that relational arguments are not equivalent to null syntactic pronouns and are present only in the semantics.

The account of relational nouns and pronouns, resumptive and otherwise, is variable-free: pronouns are functions on their antecedents and no assignment functions are used in calculating pronominal reference. The variable-free theory of anaphora is couched in Glue Semantics (Dalrymple 1999, 2001) and is compared to the related variable-free semantics of Jacobson (1999). The variable-free tradition in Glue Semantics (Glue) has arisen independently of the variable-free tradition of which Jacobson (1999) is a well-known exemplar. Not only is this kind of theoretical convergence desirable and promising, at the very least it vindicates Jacobson's assertion that the variable-free hypothesis is independent of her particular implementation. The variable-free theory of Jacobson (1999), unlike the variable-free Glue theory presented here, does not solve the resumptive puzzle – a fact that is noticed and discussed by Jacobson herself (Jacobson 1999, p. 130). The difference between the two theories in this respect is traced to differing assumptions that they make about grammatical architecture. In particular, Glue Semantics assumes a level of semantic representation, while the variable-free theory of Jacobson (1999) does not. I further discuss this difference in terms of the hypothesis of *direct compositionality*. I argue that use of a semantic representation does not necessarily violate direct compositionality. I use the grammatical architecture of Lexical Functional Grammar (LFG; Kaplan and Bresnan 1982; Bresnan 2001; Dalrymple 2001) coupled with Glue Semantics to make this point explicit. I furthermore show various implications of considering direct compositionality from a Glue perspective.

The paper is organized as follows. In Section 2, I present the resumptive puzzle in more detail, first from the perspective of variable-free semantics and then from the perspective of a variable-based theory. In Section 3, I give a brief introduction to Glue Semantics and its assumptions about semantic representation and grammatical architecture. I discuss the variable-free theory of anaphora in Glue Semantics and compare it to Jacobson's theory. A novel Glue analysis of relational nouns is presented in Section 4, with special attention paid to the bound reading. I then present a brief overview of a Glue analysis of resumptive pronouns (Asudeh 2004), in Section 5. At this point, the pieces are in place to show the solution to the resumptive puzzle (Section 6). Finally, in Section 7, direct compositionality is viewed in detail through the lens of Glue Semantics, bringing to the fore various implications and directions for further research.

2. The Trouble with Relational NPs

The problems raised by relational noun phrases, and the resumptive puzzle in particular, become evident in light of Jacobson's (1999) theory of *variable-free semantics*, which builds on previous work by herself and others (see Jacobson 1999 for further references). I will rely on Jacobson (1999) as the central, programmatic presentation of the theory.

The hypothesis of variable-free semantics is that there is no crucial use of variables in natural language semantics, and that there are no assignment functions as a result. In the variable-free treatment of Hepple (1990), which Jacobson (1999) builds on, pronouns are type $\langle e, e \rangle$ identity functions on their antecedents (setting aside agreement features):

(14) *him*; $\lambda x.x$

Although Jacobson (1999) uses lambda terms to represent functions, all the variables in these terms are *bound variables*. The lambda terms could therefore be replaced by *combinators*, which are an alternative way to name and define functions without using variables (Hindley and Seldin 1986).

Jacobson's (1999) also follows Hepple (1990) in enlarging the set of syntactic categories such that categories are marked with a record of their semantic type:

(15) If A is a syntactic category and B is a syntactic category, then A^B is a category. The semantic type of an expression of category A^B is a function from the type of B to the type of A. (Jacobson 1999, p. 129, (12))

It is important to bear in mind that the superscript category is not just a 'feature' and that a category A^C is a *distinct* category from a category A. Pronouns are lexically assigned the category NP^{NP}: they are type $\langle e, e \rangle$ functions from *NP*s to *NP*s.

The move made in (15) solves two problems that would arise from assigning pronouns the simple category *NP*. The first problem is a tension between the syntactic categories of VPs and their semantic types. Setting slash directionality aside (as does Jacobson), using the standard notation of | for an undirected slash, a VP not containing a pronoun is of category $S|NP$ and is of semantic type $\langle e, t \rangle$. A VP containing a pronoun should not be of category $S|NP$, though, because its semantic type is $\langle e, \langle e, t \rangle \rangle$ (since it has function-composed with a pronoun of type $\langle e, e \rangle$). This problem is solved through the assignment of the syntactic category NP^{NP} to pronouns. A transitive verb like *love* is of category $(S|NP)|NP$ lexically, but can shift via the operation $\mathbf{g_{CAT}}$ (the syntactic part of the Geach type shift; Jacobson 1999, p. 130), to $(S|NP)^{NP}|NP^{NP}$. Having picked up its object argument, *loves him* is of category $(S|NP)^{NP}$. This is distinct from the category of an ordinary VP, which is the unshifted $S|NP$.

The second problem with assigning pronouns the category *NP* is that it would be possible to apply the result of function-composing a transitive verb and a pronoun function to a proper name that has not been type-raised and is therefore of type *e*. The result would be that the proper name would erroneously saturate the argument slot corresponding to the pronoun. In the case of *love-him*′, a subject proper name in its base *e* type would become the lovee rather than the lover. The category assignment in (15) and the Geach shift solve this problem as well. The VP *loves him* is of category $(S|NP)^{NP}$ and cannot apply to *NP*, the non-type-raised category of a proper name. The only way for the name to combine with a category $(S|NP)^{NP}$ is to first type-raise from *NP* to $S|(S|NP)$ and then shift to $S^{NP}|(S|NP)^{NP}$ and finally to apply to the VP, yielding S^{NP}. This results in the name saturating the lover slot.

The assignment of the category NP^{NP} to pronouns together with application of the Geach shift to functors on pronouns allows Jacobson (1999) to maintain the following key generalization:

GENERALIZATION: Consider any expression C which contains no pronouns which are unbound within C, but which does contain an NP (or a pronoun bound within C). Consider further an expression C′ which is exactly like C except that C′ contains an unbound pronoun in the position of the NP (or the pronoun bound within C). Then if C can grammatically occur in some environment, so can C′. (Jacobson 1999, p. 124, (7))

Let us call this generalization *substitution with unbound pronouns*, or SWUP. It basically states that pronouns appear where full nominal expressions do. SWUP is not parochial to Jacobson's theory and is a generalization that any theory of anaphora should maintain. It is typically maintained by assigning pronouns the same category as full nominals (NP or DP, depending on the theory). Jacobson (1999) denies the assumption that SWUP entails identity of syntactic categories, but maintains the SWUP generalization itself: It still follows that an expression containing a pronoun can occur where an expression with a full NP substituted for the pronoun can occur, since an expression that can combine with a pronoun is just a shifted version of an expression that combines with a full NP.

Jacobson (1999, p. 130) notes in passing that a consequence of the variable-free treatment of pronouns using superscripted categories and the Geach rule paired with the SWUP generalization is that an apparently tidy treatment of resumptive-sensitive complementizers is available, where the class of resumptive-sensitive complementizers is taken to include what might pretheoretically be called relative pronouns (McCloskey 2002). Resumptive pronouns are the primary evidence that SWUP holds only in one direction. Although SWUP means that substituting an unbound pronoun for a full NP should always be possible (modulo agreement features), the reverse direction does not hold: SWUP does not state that an unbound pronoun can always be replaced by a full NP or bound pronoun (even setting aside binding theory). There is thus no down-shift from A^B to A corresponding to the up-shift from A to A^B. A resumptive-sensitive complementizer can then be lexically specified as requiring a right argument of category S^{NP}. Such a complementizer could not combine with a category S, which would either contain no pronouns or only bound pronouns. As an illustration, consider the following two hypothetical relative clauses and their accompanying CG derivations, where WH_{pro} is a relative pronoun requiring a constituent containing a pronoun as its complement. The subscripts on the slashes indicate direction of application (left or right):

(16)a. person WH_{pro} Mary knows him

$$\dfrac{\text{person}}{N} \quad \dfrac{\mathrm{WH}_{pro}}{(N/_{\mathrm{L}}N)/_{\mathrm{R}}S^{NP}} \quad \dfrac{\text{Mary}}{NP} \quad \dfrac{\text{knows}}{(S/_{\mathrm{L}}NP)/_{\mathrm{R}}NP} \quad \dfrac{\text{him}}{NP^{NP}}$$

$$\dfrac{NP}{S/_{\mathrm{R}}(S/_{\mathrm{L}}NP)}\,\textbf{lift} \qquad \dfrac{(S/_{\mathrm{L}}NP)/_{\mathrm{R}}NP}{(S/_{\mathrm{L}}NP)^{NP}/_{\mathrm{R}}NP^{NP}}\,\textbf{geach}$$

$$\dfrac{S/_{\mathrm{R}}(S/_{\mathrm{L}}NP)}{S^{NP}/_{\mathrm{R}}(S/_{\mathrm{L}}NP)^{NP}}\,\textbf{geach} \qquad \dfrac{(S/_{\mathrm{L}}NP)^{NP}/_{\mathrm{R}}NP^{NP} \quad NP^{NP}}{(S/_{\mathrm{L}}NP)^{NP}}$$

$$\dfrac{S^{NP}/_{\mathrm{R}}(S/_{\mathrm{L}}NP)^{NP} \quad (S/_{\mathrm{L}}NP)^{NP}}{S^{NP}} \qquad \dfrac{(N/_{\mathrm{L}}N)/_{\mathrm{R}}S^{NP} \quad S^{NP}}{(N/_{\mathrm{L}}N)} \qquad \dfrac{N \quad (N/_{\mathrm{L}}N)}{N}$$

(17)a. *person WH_{pro} Mary knows Kim

$$\dfrac{\mathrm{WH}_{pro}}{(N/_{\mathrm{L}}N)/_{\mathrm{R}}S^{NP}} \quad \dfrac{\text{Mary}}{NP} \quad \dfrac{\text{knows}}{(S/_{\mathrm{L}}NP)/_{\mathrm{R}}NP} \quad \dfrac{\text{Kim}}{NP}$$

$$\dfrac{(S/_{\mathrm{L}}NP)/_{\mathrm{R}}NP \quad NP}{(S/_{\mathrm{L}}NP)} \qquad \dfrac{NP \quad (S/_{\mathrm{L}}NP)}{S} \qquad \dfrac{(N/_{\mathrm{L}}N)/_{\mathrm{R}}S^{NP} \quad S}{\textbf{FAIL}}$$

Jacobson's theory thus has potential as a lexicalist account of resumption that localizes the availability of resumption in properties of the complementizer (whether it takes an S or S^{NP} complement). Furthermore, there is nothing special about resumptive pronouns:

they are just like the other pronouns in the theory, which meets a basic desideratum of a theory of resumption (McCloskey 2002; these points are discussed further in Section 5).

Although this analysis of resumptive-sensitive complementizers is a promising start to a theory of resumption, it has a serious shortcoming, which Jacobson herself notes (1999, pp. 129–130). Since relational nouns can be bound, like pronouns, the mechanism for pronominal binding extends to relational nouns. It follows that a relational noun such as *neighbour* is of category N^{NP} according to Jacobson (1999). A relational noun phrase is therefore an $\langle e, e\rangle$ function of category NP^{NP}, just like a pronoun.

This is where the problem arises. If a relational noun phrase is of category NP^{NP}, then a resumptive-sensitive complementizer could just as easily have its requirements met by a complement containing a relational noun. To put it another way, the theory predicts that relational nouns and resumptive pronouns should be in free variation. That is not the case in any language to my knowledge. Jacobson (1999, p. 130, fn. 10) notes the problem:

> It is unclear to me, however, whether [relational nouns] satisfy the resumptive-pronoun environments. While (ii) below is perhaps not too bad, on the whole I find examples like these quite marginal, and have no explanation as to why:
>
> (i) ?*every man who$_i$ Mary likes a neighbor$_i$/who$_i$ a neighbor$_i$ got arrested
>
> (ii) ??every state which an adjacent state has a Republican governor

The broader problem, then, is that Jacobson's theory makes false predictions about the distribution of NP^{NP} categories that are not pronouns.

These ungrammatical relativized NPs contrast with grammatical *such that* counterparts (Higginbotham 1984):

(18) Every state is such that an adjacent state has a Republican governor.

However, *such that* relatives arguably do not involve binding, unlike the restrictive relative clauses in Jacobson's (i) and (ii), and do not need even a pronominal connection to the modified NP (Pullum 1985, p. 292, (le)):

(19) The old crone had a manner such that even the children who saw her pass in the street would shudder and turn away.

A pronoun in a *such that* relative is therefore not bound and is not a resumptive pronoun (Asudeh 2004, pp. 403–407). These relatives are not relevant to the problem at hand, then.

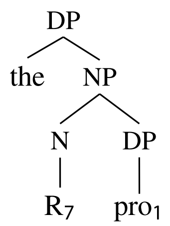
There are also two particular observational/empirical problems that should be noted. First, Jacobson seems to be operating under the assumption that *English* has resumptive pronouns. This is a controversial and uncommon position and it has been explicitly argued against quite convincingly (Chao and Sells 1983; Sells 1984). Second, my own native intuition and that of every native speaker who I have consulted is that the relativized NPs in (i) and (ii) above (or sentential versions of these and similar examples that result from tacking on an appropriate intransitive verb at the end) are not just marginally grammatical, but rather completely ungrammatical.

Jacobson's overall approach is nevertheless promising. As noted in the introduction, it seems pretheoretically desirable to account for all bound readings using the same mechanism. However, we also noted that the mechanism cannot be completely undiscriminating. This is the source of the problems discussed in this section, given that in Jacobson's theory both relational nouns and pronouns are functions on their antecedents and have the same category, NP^{NP} (Jacobson 1999). Jacobson's account serves to bring an important puzzle to the fore:

(20) If the implicit argument of a relational noun can be bound like a pronoun, why is it impossible for the argument to function resumptively?

This is the resumptive puzzle and Jacobson's theory helps to identify it, although it does not solve it.

Once we have identified the puzzle, it becomes clear that it is just as much a puzzle for variable-based theories. An extension of Heim and Kratzer (1998) variable-based semantics for E-type pronouns (Evans 1977) to relational nouns, along lines suggested by Heim and Kratzer themselves, likewise raises but does not solve the resumptive puzzle. Heim and Kratzer (1998, pp. 290–297) present an analysis, based on Cooper (1979), in which E-type pronouns are represented as follows:

(21) 

The terminal element *R* is a free, relational variable and its complement *pro* will ultimately be a bound variable. The free variable *R* receives its value from the context and the *pro* will be bound by the E-type pronoun's binder; the free and bound variables together ensure that the E-type pronoun behaves somewhat like a bound variable, but receives part of its interpretation contextually. The fact that the structure is headed by a definite determiner explains why E-type pronouns can be replaced by definite descriptions. Heim and Kratzer (1998) assume that a DP that has an unpronounced complement and that is headed by a definite determiner gets spelled out as a pronoun (also see Elbourne 2001, 2002). Thus, the entire structure in (21) will be realized as a pronoun.

Heim and Kratzer (1998, p. 291) note that the free variable *R* in the E-type pronoun is essentially like a relational noun, except that its semantic content is contextually, rather than lexically, provided. This points the way to an extension of the analysis of E-type pronouns to an analysis of bound relational nouns: assume the NP structure in (21), but substitute the relational noun's lexically specified relation for the free relational variable *R*. This NP then serves as a normal NP complement to a determiner, without restricting the D to the definite determiner. Crucially, though, there would be a null pronominal DP complement to the relational noun. There is no a priori reason to believe that such a structure is parochial to English, and it would presumably extend to any language with relational nouns, including those that have resumptive pronouns.

It is evident that the resumptive puzzle is equally problematic for the variable-based account. The analysis of relational nouns alluded to by Heim and Kratzer (1998) posits a pronominal element in the relational noun. The question becomes:

(22) Why can the null pronominal in a relational noun phrase not be used resumptively?

The resumptive puzzle is thus a genuine problem for semantic theory, since it arises in both variable-free and variable-based approaches.

Notice that it is not possible to claim that null pronominals cannot be resumptive, because the distribution of the resumptive-sensitive complementizer in Irish shows that they in fact can (McCloskey 1979, 1990, 2002). For example, inflected prepositions in Irish, are typically analyzed as having a *pro* complement (McCloskey 1979, 1990), and this null complement can function resumptively:

(23) an fear a dtabharann tú an tairgead **dó**
the man COMP give you the money to.him
the main to whom you give the money
(McCloskey 1979, 6, (3))

(24) Céacu ceann a bhfuil dúil agat **ann**?
which one COMP is liking at.you in.it
Which one do you like?
(McCloskey 2002, 189, (10b))

The mutation induced by the complementizer *a* in these examples indicates that it is the resumptive-sensitive complementizer (McCloskey 1979). In the relative clause in (23) it is the null pronominal complement of the inflected preposition *dó* ('to him') that is the resumptive. Similarly, in the question in (24) the resumptive is the null pronominal complement of the preposition *ann* ('in it'). Thus, null pronominals can be resumptives. Furthermore, since the pronominal complement of the preposition is embedded in a PP, the examples show that even deeply embedded pronouns may be resumptive.

Lastly, it should be noted that binding theory cannot account for the difference between relational NPs and pronouns with respect to resumption. The binding principle that applies to non-pronominal noun phrases is Principle C, but Principle C regards binding *of* NPs, whereas the resumptive puzzle is about binding *into* an NP. If we appeal to binding theory to explain why the implicit argument of a relational noun cannot be a resumptive (perhaps because of Principle B), then we are left with no explanation of why it can be locally bound, as in (1). The lack of a binding-theoretic solution to the resumptive puzzle is thus not dependent on a particular implementation of binding theory. It holds not just for variable-based theories that treat the relational argument as a null *pro*, but also for the variable-free treatment of Principle B offered by Jacobson (2003).

2.1. *Summary*

The variable-free account of Jacobson (1999) raises the resumptive puzzle about relational nouns. Both the variable-free account of Jacobson (1999, 2003) and an extension of the variable-based account of Heim and Kratzer (1998) fail to solve the puzzle. The puzzle is thus not idiosyncratic to variable-free theories, but is rather a deeper problem that arises from bound relational nouns. In Section 3, I introduce Glue Semantics and its implementation of variable-free anaphora. I then use the theory to build analyses of relational nouns and resumptive pronouns in Sections 4 and 5. I show in Section 6 how the Glue theory solves the resumptive puzzle.

3. An Introduction to Glue Semantics

Readers who are familiar with Glue Semantics may wish to skip ahead to Section 3.1 and refer as needed to Appendix A, which presents the Glue logic and proof rules and term assignments for the fragment of linear logic it assumes. Readers who seek a fuller introduction should refer to Dalrymple (2001) and the papers in Dalrymple (1999), especially Dalrymple et al. (1999a,b,c).

In Glue Semantics, *meaning constructors* for semantic composition are obtained from lexical items instantiated in particular syntactic structures. Each constructor has the following form:

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

$\mathcal{M}$ is a term from some representation of meaning, a *meaning language*, and G is a term of the Glue logic that sticks meanings together, i.e., performs composition. The colon is an uninterpreted pairing symbol. Linear logic (Girard 1987), or more precisely a fragment of linear logic, serves as the Glue logic (Dalrymple et al. 1993, 1999a,b). The meaning constructors are used as premises in a (linear logic) proof that consumes the lexical premises to produce a sentential meaning. A successful Glue proof proves a conclusion of the following form (following Crouch and van Genabith 2000, p. 117), where G_t is a term of type t:[2]

(26) $\Gamma \vdash \mathcal{M} : G_t$

[2] Proof goals in Glue are discussed in more detail in Asudeh (2004, pp. 86–87).

Semantic ambiguity (e.g., scope ambiguity) results when there are alternative derivations from the same set of premises. The logics for $\mathcal{M}, G$ are presented in Appendix A.1.

Asudeh (2004, p. 71ff.) discusses linear logic in detail from the perspective of substructural logics and compares Glue Semantics to related type-logical approaches to Categorial Grammar (Morrill 1994; Carpenter 1997; Moortgat 1997). The key difference between Glue and Categorial Grammar is that the latter rejects a level of syntax that is separate from the syntax of semantic composition whereas the former accepts such a level. The acceptance of a separate level of syntax allows the Glue logic for semantic composition to be commutative, permitting reordering of premises, without wildly overgenerating. What is particularly important for present purposes is that linear logic lacks the structural rules of contraction and weakening and is therefore a resource logic, unlike classical and intuitionistic logics. All premises must be used in a linear logic proof and no premises may be reused. Let us call this kind of resource accounting *resource sensitivity*. The following comparison to classical/intuitionistic logic serves as an illustration of this (Asudeh and Crouch 2002c; $\multimap$ is linear implication and $\otimes$ is one form of linear conjunction, multiplicative conjunction):

(27)

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

(28)

Premise non-use	
Classical/Intuitionistic Logic	Linear Logic
$A, B \vdash A$	$A, B \nvdash A$
Can ignore premise B	Cannot ignore premise B

Resource sensitivity tightly constrains the proof space of linear logic. More importantly from a linguistic perspective, the resource sensitivity of linear logic models the resource sensitivity of natural language semantics, whereby each meaningful element makes its meaning contribution exactly once (cf., for example, the *bounded closure* of Klein and Sag 1985), as explored in detail by Asudeh (2004). Thus, resource sensitivity constrains derivations in linguistically desirable ways and will be shown to have key theoretical consequences in Sections 3.1, 4, and 5.

The fragment of linear logic I assume is the modality-free, multiplicative fragment of intuitionistic linear logic, MILL. It is not a strictly propositional logic, because it has universal quantification, but it is not fully higher order, since the quantification is strictly limited to universal quantification over t-type atoms of the linear logic (Crouch and van Genabith 2000, p. 124).[3] The logic MILL lacks existential quantification and negation. It is therefore quite weak from a proof-theoretic perspective (there are many things it cannot prove), but it is strong enough for central concerns of linguistic semantics, such as basic composition of functors and arguments, anaphora, and scope. See Appendix A for further details.

This paper particularly uses three proof rules of this fragment of linear logic. In a natural deduction presentation, these are conjunction elimination for $\otimes$ and implication introduction and elimination for $\multimap$ (a.k.a. 'abstraction' or 'hypothetical reasoning' for implication introduction and 'modus ponens' for elimination), as shown in (29).

(29) **Implication Elimination** **Implication Introduction**

$$\frac{\begin{array}{c}\vdots\\ A\end{array} \qquad \begin{array}{c}\vdots\\ A \multimap B\end{array}}{B}\multimap_{\mathcal{E}} \qquad\qquad \frac{\begin{array}{c}[A]^1\\ \vdots\\ B\end{array}}{A \multimap B}\multimap_{\mathcal{I},1}$$

Conjunction Elimination

$$\frac{\begin{array}{c}\vdots\\ A \otimes B\end{array} \qquad \begin{array}{c}[A]^1\ [B]^2\\ \vdots\\ C\end{array}}{C}\otimes_{\mathcal{E},1,2}$$

A premise in brackets with a numerical flag indicates an assumption; the flags keep track of which assumptions have been withdrawn and

[3] Kokkonidis (2003, 2004) defines a version of Glue Semantics that explicitly limits quantification to first-order; the treatment is extremely promising, but at this point not well-understood or widely adopted in Glue analyses.

which are active. The rules for implication may be familiar from classical and intuitionistic logic and the rule for $\otimes$ might be too, except that it looks similar to the rule for discharging *disjunction*. The resource perspective on linear logic can make the intuition behind this apparently puzzling similarity clear. Just as in classical or intuitionistic logic we can only be sure that a disjunction is true or provable if both disjuncts can be used to establish some conclusion (given that we do not know *which* disjunct confirms the disjunction), in linear logic we can only be sure that we can use a multiplicative conjunction of two resources if we know that both resources can be used independently.

The linear logic proof rules construct proof terms via the Curry–Howard isomorphism (a.k.a. "formulas-as-types"; Curry and Feys 1958, 1995; Howard 1980), which establishes a formal correspondence between natural deduction and terms in the lambda calculus. These terms reconstruct the proofs in corresponding types. One useful application of the terms is in stating identity criteria for proofs, so that we know when two proofs are equivalent and when they are not; thus, term reduction is related to proof normalization (Gallier 1995). The basic insight behind the isomorphism is that implications correspond to functional types, so that implication elimination corresponds to *functional application* and implication introduction corresponds to *abstraction*. The basic isomorphism, discovered by (Curry and Feys 1958; Curry and Feys 1995 contains the most relevant sections), was extended to deal with various other types by Howard (1980). The Curry–Howard term assignments for the three rules in (29) are:

(30) Application : Impl. Elim.

$$\dfrac{\begin{matrix}\vdots & \vdots\\ a : A & f : A \multimap B\end{matrix}}{f(a) : B}\multimap_{\mathcal{E}}$$

Abstraction : Impl. Intro.

$$\dfrac{\begin{matrix}[x : A]^1\\ \vdots\\ f : B\end{matrix}}{\lambda x.f : A \multimap B}\multimap_{\mathcal{I},1}$$

Pairwise substitution : Conj. Elim.

$$\dfrac{\begin{matrix} & [x : A]^1\ [y : B]^2\\ \vdots & \vdots\\ a : A \otimes B & f : C\end{matrix}}{\mathsf{let}\ a\ \mathsf{be}\ x \times y\ \mathsf{in}\ f : C}\otimes_{\mathcal{E},1,2}$$

As noted above, implication elimination corresponds to functional application, and implication introduction corresponds to abstraction.

The assumed premise in the introduction rule is associated with a variable that is abstracted over when the assumption is discharged. The term constructor let is possibly less familiar. A multiplicative conjunction $A \otimes B$ corresponds to a tensor product $a \times b$, where a is the proof term of A and b is the proof term of B (see the rule for conjunction introduction ($\otimes_{\mathcal{I}}$) in Appendix A.3.). However, let prevents projection into the individual elements of the tensor pair and therefore enforces pairwise substitution (Abramsky 1993, Benton et al. 1993; Crouch and van Genabith 2000, p. 88), such that a let expression β-reduces as follows:

(31) $\text{let } a \times b \text{ be } x \times y \text{ in } f \Rightarrow_{\beta} f[a/x, b/y]$

The substitution of the pair is simultaneous and does not involve projection into the members. So let is not forbidding and is just a slightly more structured form of functional application.

It is the Curry–Howard term assignments that determine operations in the meaning language. I use the locution "operations in the meaning language" purposefully. The term assignments constructed by rules of proof for linear logic result in *linear* lambdas (Abramsky 1993); these are lambda terms in which every lambda-bound variable occurs exactly once (i.e., no vacuous abstraction and no multiple abstraction). The proof terms therefore satisfy resource sensitivity. However, lexically contributed meanings need not contain only linear lambdas (for a similar point about the Lambek Calculus, see Moortgat 1997, 122ff.). This is *not* a violation of the isomorphism though, because the isomorphism says nothing about the internal structure of the functions that it constrains in correspondence to the rules of proof. Thus, the isomorphism constructs proof terms that are linear, but the proof terms are not identical to the meaning language, although they are responsible for the operations in the meaning language that correspond to rules of proof. In summary, the meaning language needs to support operations determined by the Curry–Howard for the three rules in (30).

The meaning language therefore needs to minimally support a notion of application and abstraction, as well as product pairs for the multiplicative conjunction (see Appendix A.1). Work in Glue Semantics has traditionally assumed that the meaning language is a lambda calculus of some kind. The meaning language can be construed as simply being a convenient representation for what is in fact the model theory itself, just as discussed by Jacobson (1999, p. 122).

The lambda calculus is one convenient way to describe the functions that are actually in the models, but it is not the only one. For example, the meaning constructor in (32) could instead be represented as (33):

(32) $\lambda x.comedian(x) : a \multimap b$

(33) $\lambda^{*}x.\mathbf{comedian}\ x : a \multimap b$

The meaning language side of (33) uses abstraction in combinatory logic (Curry and Feys 1958), where $\lambda^{*}.M$ is not part of the formal system of terms, but is rather part of the metatheory and is constructed from the combinators S and K and parts of M (Hindley and Seldin 1986, pp. 25–28). The possibility of using combinators underscores the fact that the meaning language for Glue is variable-free in the same sense as Jacobson's theory: there is no *crucial* use of variables, since the variables are bound. Even implication introduction, with the apparently free variable in the assumed premise, does not pose a problem, because this rule just corresponds to abstraction and we have just seen that abstraction can be defined in terms of combinators. The meaning language is presented in Appendix A.1. I assume a simple extensional semantics, as the main area of concern is semantic composition, rather than truth conditions per se.

There are three further proof rules for MILL: conjunction introduction ($\otimes_{\mathcal{I}}$) and universal introduction and elimination ($\forall_{\mathcal{I}}$ and $\forall_{\mathcal{E}}$). These rules are given in the appendix since they complete the logic, but the first two are not used, since conjunctions and universals are not introduced in proofs (except for a single use of $\otimes_{\mathcal{I}}$ in (98), a sketch of a failed proof). They occur only in lexical specifications of, respectively, anaphoric elements and scopal elements.

The universal quantifier $\forall$ that is used in the analysis of scope occurs only in the linear logic side, G, of meaning constructors $\mathcal{M} : G$. It is important to realize that $\forall$ means *any* not *all* in linear logic (Crouch and van Genabith 2000, p. 89). Consider this from the perspective of linear logic as a resource logic. If all the resources quantified over were selected, there could be massive resource failure, since they would all be consumed in one fell swoop. Rather, the way to reason about it is that if some property holds of all such resources, then you can pick any one and know that the property holds over that one. A contrast with the existential quantifier (which is absent in this fragment) serves to highlight the fact that, despite the resource-sensitive interpretation of $\forall$, it is truly universal quantification. In the case of the existential quantifier, we know that the property holds of some resource, but we cannot arbitrarily pick *any* resource and be sure that the property hold of *that* resource. Thus, the universal quantifier in linear logic really is a universal and should be represented as such, despite its 'any' interpretation and the possible danger of overloading the symbol $\forall$. The potential overload is not a real danger here, because $\forall$ will only be used in the linear logic. It will never appear in the terms of the meaning language, where all quantifiers are represented using a functional generalized quantifier notation that is discussed further below. That the linear logic universal is a true universal is further underscored by the fact that its introduction and elimination rules are identical to those of intuitionistic logic. Since elimination of a universal is trivial, the universal elimination rule is used only implicitly.

I noted above that the Glue meaning constructors are instantiated relative to a particular syntactic parse and that it was the assumption of a syntax separate from the syntax of the proof theory that allows the logic of composition to be commutative. Glue is not necessarily bound to any particular syntactic framework,[4] but most work in Glue has been done with a Lexical Functional Grammar syntax (Kaplan and Bresnan 1982; Bresnan 2001; Dalrymple 2001). This paper also assumes an LFG syntax, but more importantly it assumes the LFG grammatical architecture, which is a *projection architecture*: different grammatical components are represented in separate modules and related by *projection functions* which map elements of one module to elements of another (Kaplan 1987, 1989). The particular architecture I assume is shown in (34), where ϕ and σ are projection functions:

(34) constituent structure $\xrightarrow{\phi}$ functional structure $\xrightarrow{\sigma}$ semantic structure

This projection architecture is what ultimately solves the resumptive puzzle while allowing a version of Glue Semantics that meets the requirements of direct compositionality.

[4] For example, Frank and van Genabith (2001) present Glue Semantics for Lexicalized Tree-adjoining Grammar and Asudeh and Crouch (2002c) present an HPSG version.

The linear logic resources used for semantic composition in Glue-LFG are node labels in semantic structure (s-structure), instantiated by the σ projection function, which maps functional structure (f-structure) nodes to s-structure nodes (Dalrymple et al. 1993, 1999b; Dalrymple 2001). This means that the meaning constructors contributed by lexical items are instantiated by σ projections on f-structure equations. These f-structure equations are standardly called f-descriptions, since they describe functional structures. Let us call σ-mapped f-structure equations 's-descriptions', since they describe semantic structures. Meaning constructors are instantiated by s-descriptions. For example, the proper name *Mary* provides the meaning constructor in (35a) and the intransitive verb *laughed* the one in (35b).

(35)a. $mary : \uparrow_{\sigma_e}$
b. $\lambda x.laugh(x) : (\uparrow \text{SUBJ})_{\sigma_e} \multimap \uparrow_{\sigma_t}$

The σ-projections of the f-descriptions get instantiated in a parse. For example, if we had the f-structure (36), with node labels as indicated, then the f-descriptions in (35) would get instantiated as in (37):

(36) $$f\begin{bmatrix} \text{PRED} & \text{'laugh'} \\ \text{SUBJ} & g\begin{bmatrix} \text{PRED} & \text{'Mary'} \end{bmatrix} \end{bmatrix}$$

(37)a. $mary : g_{\sigma_e}$
b. $\lambda x.laugh(x) : g_{\sigma_e} \multimap f_{\sigma_t}$

The lexical item *Mary* contributes the resource that is the σ-projection of its f-structure (the latter indicated as usual by $\uparrow$); similarly, the lexical item *laughed* contributes a resource that is an implication from the σ-projection of its subject to the σ-projection of the verb, where $(f\ \text{SUBJ}) = g$ in (36). However, it is standard practice in Glue work to name meaning constructors mnemonically and to suppress the σ-projection and type subscripts where convenient. Therefore, the normal abbreviation for the resources contributed by *Mary* and *laughed*, when the former is the subject of the latter, would be m and $m \multimap l$. This naming convention allows a schematic presentation of meaning constructors that abstracts away from how they are derived from the syntax, focusing instead on the compositional semantics. I will call meaning constructors written in terms of s-descriptions *generalized meaning constructors* and those written using the mnemonic convention *schematic meaning constructors*. I will also present lambda terms in their η-reduced form where convenient; for example, $\lambda x.laugh(x) \Rightarrow_\eta laugh$.

3.1. *Anaphora in Glue Semantics*

Anaphora resolution in Glue Semantics has been variable-free from the start (Dalrymple et al. 1999c), and independently of the variable-free tradition in Categorial Grammar. This can ultimately be traced to the commutativity of the Glue logic. In the CG tradition discussed above, the pronoun is a function on its antecedent but cannot combine with it directly, since the pronoun does not occur adjacent to its antecedent in the string and the non-commutative logic of CG does not allow arbitrary reordering of premises to permit direct application. This necessitates a series of function compositions such that a function that has composed with the pronoun applies to the antecedent. In contrast, since the Glue logic is commutative, the pronoun can directly apply to its antecedent. Given the possibility of such application, there is no temptation to use assignment functions for pronouns, since a cleaner alternative is immediately apparent.

A pronoun in this Glue tradition is of the same syntactic category as a full non-pronominal noun phrase, whether it is NP or DP (I assume NP, but nothing hinges on this). Therefore, the SWUP generalization follows: pronominal NPs and non-pronominal NPs have the same distribution, because they have the same category. This is just how SWUP is captured in most theories. A pronoun has a meaning constructor that makes crucial use of multiplicative conjunction ($\otimes$), as shown here:

(38) $\lambda z.\ z \times z : (\uparrow_\sigma \text{ANTECEDENT})_e \multimap ((\uparrow_\sigma \text{ANTECEDENT})_e \otimes \uparrow_{\sigma_e})$

A schematic representation of the pronoun's meaning constructor is as follows, where A is the antecedent's resource and P is the pronoun's resource:

(39) $A \multimap (A \otimes P)$

The pronoun's meaning constructor consumes its antecedent's resource to produce a conjunction of the antecedent resource and the pronoun's resource. The pronoun has a functional type from type e to the product type $e \times e$. The pronoun's type is therefore $\langle e, \langle e \times e \rangle\rangle$. The possible values of ANTECEDENT at s-structure are constrained by syntactic factors (Dalrymple et al. 1999c, p. 58), including LFG's binding theory, which is stated using f-structural relations and the mapping from functional structure to semantic structure (Dalrymple 1993; Bresnan 2001).

We can construct the proof in (41) for the simple example in (40), using the mnemonic convention for naming resources, where p indicates 'pronoun' ($\Rightarrow_\beta$ indicates β-reduction of a lambda term).

(40) Bo fooled himself.

(41)

$$\dfrac{\dfrac{\dfrac{bo : b \qquad \begin{array}{c}\lambda z.z \times z :\\ b \multimap (b \otimes p)\end{array}}{bo \times bo : b \otimes p}\multimap_{\mathcal{E}} \qquad \dfrac{\dfrac{[x : b]^1 \qquad \begin{array}{c}\lambda u \lambda v.\mathit{fool}(u, v) :\\ b \multimap p \multimap f\end{array}}{\lambda v.\mathit{fool}(x, v) : p \multimap f}\multimap_{\mathcal{E}} \qquad [y : p]^2}{\mathit{fool}(x, y) : f}\multimap_{\mathcal{E}}}{\text{let } bo \times bo \text{ be } x \times y \text{ in } \mathit{fool}(x, y) : f}\otimes_{\mathcal{E},1,2}}{\mathit{fool}(bo, bo) : f}\Rightarrow_\beta$$

Note that there is nothing special about the transitive verb *fool*. It has not undergone a type shift or been modified in any way to accommodate the pronoun. Note also that the resource corresponding to the pronoun is the right member of the conjunction pairing and that it is a type e atomic resource, just like that of a name. However, the proof rule for conjunction elimination requires simultaneous substitution of the products and does not permit separate projection into one or the other. Finally, observe that the pronoun does not correspond to a free variable, since the corresponding variable is lambda-bound. Thus, we have a variable-free analysis of pronouns.

The equivalence to Jacobson's (1999) analysis may not be immediately clear, since in her analysis pronouns are type $\langle e, e\rangle$ functions and in Glue they are type $\langle e, \langle e \times e\rangle\rangle$ functions. Things become clearer if we take resource sensitivity into account. On the Glue account, the pronoun takes its binder as an argument directly. In doing so, it must replicate its antecedent's meaning, otherwise there will be resource failure elsewhere in the derivation, as the binder's resource must be available for composition with some functor. In Jacobson's account, the pronoun does not directly apply to its antecedent; instead, it undergoes function composition with an adjacent functor and after successive function compositions it eventually gets bound off by a binder. Although Categorial Grammar, like Glue, is resource-sensitive (Moortgat 1997), the fact that the pronoun never directly takes its binder as an argument means that the binder is available to be taken as an argument by the functor that contains the pronoun.

The proof (41) serves to illustrate the Curry–Howard correspondence between the linear logic and the meaning language. As noted above, a consequence of the Curry-Howard isomorphism is that meaning terms cannot constrain proofs. It is therefore sufficient to show proofs using only the linear logic, since the meaning terms follow. In practice, it becomes useful to show the meaning terms for more complicated proofs. However, the proof trees then become ungainly. I therefore adopt the convention of showing meaning composition in a separate list-style proof. Not only is this useful from a presentational point of view, it also underscores on the one hand that linear logic terms are the engine of the theory and on the other hand that the proofs are abstract objects that can be written down in various ways.

In summary, the Glue account of anaphora is variable-free. Due to the mechanics of the Glue account versus Jacobson's variable-free CG account, pronouns in the former are of type $\langle e, \langle e \times e \rangle \rangle$, as opposed to being of type $\langle e, e \rangle$. The SWUP generalization is captured in the syntax, since pronouns and non-pronominal NPs have the same category. The pronoun's antecedent is represented at semantic structure via the feature ANTECEDENT, which will often be abbreviated as ANT. Syntactic restrictions can be placed on pronominal binding, as will become clear in Section 4.1.

3.2. *Summary*

This has been a necessarily concise overview of Glue Semantics. The key points to remember are:

1. Glue Semantics uses linear logic to compose lexically contributed meanings instantiated in a syntactic parse.
   (a) Linear logic is a resource logic and is therefore resource-sensitive: resources cannot be freely discarded or reused.
   (b) The meaning terms are related to the linear logic via the Curry–Howard isomorphism; this ensures that success or failure of proof is due solely to the linear logic.
2. The theory of anaphora in Glue Semantics is variable-free. Pronouns and other anaphoric elements are functions on their antecedents.
   (a) These functions are constructed using the multiplicative conjunction. This has the effect that the antecedent's resource

is consumed in calculating anaphoric reference, but replicated so that it can be used elsewhere as required.

3. Glue-LFG assumes a parallel projection architecture in which projection functions relate separate grammatical modules.

I will spell out further details as required below. The logic is presented in full in Appendix A.

4. Relational Nouns

In this section I offer an analysis of relational nouns that builds on the variable-free, product type analysis of pronouns. I take noun phrases headed by relational nouns as central representatives of the broader class of relational noun phrases discussed in Sections 1 and 2 (in addition to those headed by relational nouns, those modified by a relational adjective and those that are contextually relational). Relational nouns have an internal argument and are therefore of type $\langle e, \langle e, t\rangle\rangle$, rather than of type $\langle e, t\rangle$ like common nouns. I begin with a brief presentation of quantifier-noun composition in Glue. I then discuss circumstances in which the implicit argument of a relational noun is not externally bound. In Section 4.1, the bound reading is then considered in detail, with particular attention paid to semantic composition, scope, reconstruction, and crossover. Section 4.1.2 discusses interactions between the bound reading of relational nouns and overt realizations of the relational argument in genitives (*Kim's mother*) and obliques (*the mother of Kim*). Section 4.1.3 discusses licit binding of the relational noun's implicit argument vs. illicit binding of entire relational NPs.

The schematic meaning constructor for an ordinary common noun is as follows, where the λ-term can naturally be η-reduced:

(42) $\lambda x.clown(x) : v \multimap r$

The terms v and r stand for the s-structure attributes VAR(IABLE) (type e) and RESTR(ICTION) (type t).

Assuming a theory of generalized quantifiers, a noun serves as the restriction of a type $\langle\langle e, t\rangle, \langle\langle e, t\rangle, t\rangle\rangle$ determiner. There are a variety of ways to represent quantifiers on the left hand side of the meaning constructor, depending partly on the logic chosen for the meaning language. I opt for the three-place functional representation (43).

(43) $\lambda R\lambda S.no(x,\ R(x),\ S(x))$

The meaning constructor for a quantifier consumes its restriction and looks for any $\langle e, t\rangle$ implication that depends on the resource of the quantified NP:

(44) $\lambda R \lambda S.no(x, R(x), S(x))$:
$(v_e \multimap r_t) \multimap \forall X.[((\text{SPEC} \uparrow)_{\sigma_e} \multimap X_t) \multimap X_t]$

The fact that we universally quantify over X allows the quantifier to take higher scope by introducing a hypothesis on the resource $(\text{SPEC} \uparrow)\sigma$ that corresponds to the determiner's noun phrase, discharging the dependency on this resource locally, and then reintroducing it at a later point in the derivation. Notice that the universally quantified subformula in the Glue logic is what all quantifiers have as the expression for finding their nuclear scope and says nothing about the semantics of the determiner in question, which is represented as a functional quantifier in the meaning language. The reader is referred to Dalrymple et al. (1999c), Dalrymple (2001), and Crouch and van Genabith (1999) for further details.

Using the mnemonic convention described above and suppressing σ and type subscripts, we get the lexically contributed premises in (46) for sentence (45):

(45) No clown laughed.

(46)

1.	$\lambda R \lambda S.no(x, R(x), S(x))$: $(v \multimap r) \multimap \forall X.[(c \multimap X) \multimap X]$	Lex. **no**
2.	$clown : v \multimap r$	Lex. **clown**
3.	$laugh : c \multimap l$	Lex. **laughed**

From these premises we construct the proof in (47).

(47)
$$\dfrac{\dfrac{\begin{array}{l}\lambda R \lambda S.no(x, R(x), S(x)) :\\ (v \multimap r) \multimap \forall X.[(c \multimap X) \multimap X]\end{array} \quad clown : v \multimap r}{\begin{array}{l}\lambda S.no(x, clown(x), S(x)) :\\ \forall X.[(c \multimap X) \multimap X]\end{array}} \multimap_{\mathcal{E}} \qquad laugh : c \multimap l}{no(x, clown(x), laugh(x)) : l} \multimap_{\mathcal{E}}, [l/X]$$

The quantifier takes its scope by finding an appropriate dependency and consuming it through implication elimination. Note that a step of universal elimination is implicit; rather than carrying it out explicitly, the implication elimination step is annotated appropriately, since universal elimination is straightforward. The β-reduction on λS has also been carried out implicitly.

A relational noun is on the one hand like a non-relational noun, since it combines with the same determiners, but is also like a pronoun, since it has an argument that can be bound externally. However, the relational argument is not necessarily externally bound, in at least four circumstances:

1. The relational noun is type-shifted such that the argument is bound by an existential quantifier.

(48) Kim heard a rumour downtown.

2. The argument is saturated by the speaker or hearer indices.

(49) The police interviewed a neighbour who stuck up for us.

3. The argument is realized as an oblique complement of the relational noun.

(50) Kim heard a rumour about Sandy.

4. The argument is saturated in composition with a genitive NP that is the specifier or complement of the relational noun. The genitive NP has a relational argument that the relational noun saturates.

(51) Kim's neighbour is Sandy.

(52) Sandy is a neighbour of Kim's

It will be useful to discuss these circumstances before returning to the analysis of relational nouns per se. A brief note on terminology, though: a genitive complement *is* an oblique argument, but in this context I reserve the term oblique argument for arguments like *of Kim*, using the more specific terms 'genitive complement' or 'postnominal genitive' for arguments like *of Kim's*.[5]

The first case of the relational argument not being externally bound is where the argument is unspecified and we type-shift the relational noun to an ordinary common noun by existentially quantifying over the unspecified argument (Partee and Borschev 1998a,b). This is shown by the rough semantics in (54) for the example in (53):

[5] Following Partee (1983, 1997), I use the term genitive a little loosely to cover both prenominal and postnominal genitives, even though there is a category distinction between them (*my book*/* *mine book* versus * *book of my*/*book of mine*).

(53) The parents arrived late.

(54) *arrive-late* $(\iota x[\exists y[parent^*(x, y)]])$

In example (53), it is not specified whose parents arrived late, but is part of the meaning of the noun *parent* that a parent must be a parent *of* someone, which is captured through the existential quantification. Notice that *parent** is the denotation of the plural noun *parents* and that $parent(x, y)$ means that x bears the parent relation to y; relational nouns will in general be written with the nominal variable first and the relational argument second.

There are differences between relational nouns as to how readily they accept this type-shift. Consider a relational noun like *neighbour* versus relational nouns like *rumour* and *photograph*; the following sentences show the contrast that is at issue:

(55) **[Hermit A to Hermit B]**
John saw a neighbour downtown.
#It wasn't his neighbour, though.

(56)a. John heard a rumour downtown.
It wasn't a rumour about him, though.
b. John saw a photograph downtown.
It wasn't a photograph of him, though.

The additional context that sentence (55) is uttered by one hermit to another is there to preclude saturation of the relational noun's argument by the speaker or hearer index (since hermits by definition do not have neighbours; never mind why we have two such gregarious hermits). Given this context, the discourse continuation in (55) shows that it is not a presupposition that the neighbour in question is John's neighbour, because this is not cancellable. The contrast is that the relational noun *neighbour* resists the existential quantification type-shift, whereas *photograph* and *rumour* accept it.

In the hermit example saturation by the deictic speaker or hearer indices was carefully controlled for, but there are also telling contrasts between such deictic readings and the existential type-shift just described. Consider the following contrasts:

(57) The meeting was frustrating, because the parents arrived late.

(58) The meeting was frustrating, because the neighbours arrived late.

Example (58) has a natural reading in which the neighbours are the speaker's neighbours, and perhaps the hearer's too (but not necessarily: consider if I tell you about the Tenant's Association meeting in my building). It does not have a natural reading where the neighbours are just somebody or other's neighbours. Example (57) shows just the reverse effect: the most natural reading is one where the parents in questions are *not* the speaker or hearer's parents. That reading would be most naturally captured by using a possessive pronoun (i.e., *my*, *your*, or *our*). This contrast is explained if we allow the existential quantification type-shift for the first example but not the second, in which case the relational noun *neighbour*'s argument would have to function like a bound variable, which could be saturated by the speaker or hearer indices. Let us call nouns such as *neighbour* which resist the existential quantification type shift *strongly relational nouns* (SRNs) and nouns such as *parent* which allow the type shift *weakly relational nouns* (WRNs). I leave it as an open research question at this point whether the contrast between SRNs and WRNs should be lexically encoded or tied to context update.

The third circumstance in which the relational noun's argument is not externally bound is when the argument is realized as an oblique complement to the relational noun. In LFG terms, such an argument would bear an OBLIQUE grammatical function at f-structure. This subcase of relational nouns is discussed by Dalrymple et al. (1999c). An example is:

(59) Kim heard a rumour about Sandy.

Certain relational nouns, including strongly relational ones, tend to resist a non-genitive oblique argument, preferring to take the argument as a genitive oblique or specifier, as shown in (60). However, the effect is ameliorated if the oblique is sufficiently unwieldy that it would be awkward as a prenominal genitive, as shown in (61).

(60)a. ? Kim is a neighbour of Sandy.
b. Kim is a neighbour of Sandy's.
c. Kim is Sandy's neighbour.

(61)a. Kim is a neighbour of a rather quirky elderly couple who are into classic cars.

b. ? Kim is a rather quirky elderly couple who are into classic cars's neighbour.

I offer no explanation for this effect here, but I do take it as evidence that SRNs should not be disallowed from taking an oblique argument, whatever further restrictions there are.

The fourth and final circumstance to consider is when the relational noun occurs with a genitive noun phrase, where the latter can be prenominal as in (62) or postnominal as in (63):

(62) Kim's sister arrived late.

(63) A sister of Kim's arrived late.

I will assume Partee's (1983, 1997) analysis of the genitive for the sake of argument, although this is an ongoing area of research for Partee (see for example Partee and Borschev 1998b, 2003) and there are many other recent treatments that offer alternatives (among others, Jensen and Vikner 1994; Barker 1995; Vikner and Jensen 2002; Storto 2003). Genitives are not my main concern here and I think that it should suffice to pick one simple, well-known treatment.

On Partee's (1983, 1997) analysis, a prenominal genitive has as one option the semantics in (64) and a postnominal genitive has as one option the semantics in (66):

(64) $\textbf{Kim's}_{pre} = \lambda R \iota x[R(x, kim)]$

(65) $\textbf{Kim's}_{post} = \lambda R \lambda x[R(x, kim)]$

The prenominal genitive has a built-in definite article, represented with the ι operator, whereas the postnominal genitive does not (they have different types as a result). The genitive involves a relation R, one argument of which is provided by the host of the genitive affix. If the genitive NP is the specifier or complement of a relational noun, the relation is typically that of the relational noun ("inherent R"), with the genitive NP serving as the relational argument. The derivation for sentence (62) is sketched in (66):

(66)
$$\dfrac{\mathit{arrive\text{-}late} \qquad \dfrac{\dfrac{\dfrac{\lambda R\iota x[R(kim)(x)] \qquad \lambda z\lambda y.sister(z)(y)}{\iota x[(\lambda z\lambda y.sister(z)(y))(kim)(x)]}}{\iota x[(\lambda y.sister(kim)(y))(x)]}}{\iota x[sister(kim)(x)]}}{\mathit{arrive\text{-}late}(\iota x[sister(x, kim)])}$$

The inherent R genitive thus selects for a type $\langle e, \langle e, t\rangle\rangle$ argument, which is provided by a relational noun.

The genitive's relation can alternatively be contextually specified ("free R"), in which case the relational noun's relation does not saturate the genitive relation. Partee (1983, 1997), building on Stockwell et al. (1973), presents a slightly different analysis for such genitive NPs, in which they take an $\langle e, t\rangle$ argument rather than a relational argument. The $\langle e, t\rangle$ argument is provided by a non-relational noun or a relational noun that has a saturated relational argument. We thus get the corresponding versions of (64) and (65) with a contextually contributed relation R_i:

(67) $\textbf{Kim's}_{pre/free} = \lambda R_i \lambda P \iota x[P(x) \wedge R_i(x, kim)]$

(68) $\textbf{Kim's}_{post/free} = \lambda R_i \lambda P \lambda x[P(x) \wedge R_i(x, kim)]$

The following examples illustrate these free R genitives:

(69) Kim's horse won the race.

(70) The winner was Kim's horse.

In (69) and (70), the horse in question might be the horse that Kim owns, the horse that Kim bet on, the horse that Kim rode, and so on. The derivation of an abbreviated version of (69) is sketched here:

(71)
$$\dfrac{\dfrac{win \qquad \dfrac{horse \qquad \dfrac{\lambda R_i \lambda P \iota x[P(x) \wedge R_i(x, kim)] \qquad [Q]^1}{\lambda P \iota x[P(x) \wedge Q(x, kim)]}}{\iota x[horse(x) \wedge Q(x, kim)]}}{win(\iota x[horse(x) \wedge Q(x, kim)])}}{\lambda Q.win(\iota x[horse(x) \wedge Q(x, kim)])} \; {-1}$$

It may seem counterintuitive to refer to a lambda-bound R_i as "free", and indeed Partee (1983, 1997) has an unbound relational variable R_i here. However, given that I am operating under the variable-free hypothesis, there are no unbound variables: even contextual (discourse) variables must be bound (see the discussion in Jacobson (1999, p. 134) of free pronouns in sentences like *He left*).

The distinction between inherent R and free R genitives is important because they interact differently with the bound reading of relational nouns. In particular, a relational noun can only have a bound reading in the presence of a genitive if it is a free R genitive. An inherent R genitive saturates the relational argument, so it is not available for binding. This is illustrated by the following examples:

(72) Every celebrity was flattered by Warhol's portrait.

(73) Every suburbanite knows Kim's neighbour.

Example (72) contains an instance of a free R genitive: Warhol is the creator of the portrait rather than the relational argument of *portrait*, which would be the entity portrayed. This sentence does have a bound reading in which every celebrity was flattered by Warhol's portrait of the celebrity in question. In contrast, example (73) does not readily allow a free R interpretation of the genitive and the inherent R interpretation precludes a bound reading. These cases are discussed in more detail below, particularly in Section 4.1.2.

In the absence of any of the four circumstances just described, a relational noun's argument is bound by some NP external to the relational noun. This was the case in the hermit example (55), where the SRN *neighbour* occurs without a genitive or oblique and as a result its relational argument gets bound by the subject *John*, which is the only remaining way to deal with the relational argument, since SRNs resist the existential quantification type shift. The bound reading is shown even more forcefully by (1):

(1) Every suburbanite knows a neighbour.

The natural reading of this sentence is one where *every suburbanite* binds *neighbour* such that every suburbanite knows a neighbour of his/her own.

The bound reading disappears when the relational NP has an oblique argument or pre- or postnominal inherent R genitive:

(74) Every suburbanite knows a neighbour of a quirky elderly couple.

(75) Every suburbanite knows John's neighbour.

(76) Every suburbanite knows a neighbour of John's.

None of these sentences has a reading in which for every suburbanite X, there is a Y such that Y is the neighbour of X and X knows Y. This follows from the fact that the oblique or inherent R genitive either saturates the relational argument or takes the entire $\langle e, \langle e, t \rangle \rangle$ relational noun as an argument; in either case, there is no argument that can be bound externally.

Relational nouns can also undergo discourse binding, whether they are strongly or weakly relational. We can see this in relation to the following discourses:

(77) **A**: How was your baby cousin's birthday party?
B: Oh, I had a blast. The parents looked pretty bored, though.

(78) **A**: How was your baby cousin's birthday party?
B: Oh, I had a blast. The neighbours looked pretty bored, though.

These discourses are both ambiguous, but both have readings on which the relational noun's argument is discourse bound. For (77) this is the reading where the parents in question are the parents of the baby cousin. Similarly, for (78) it is the reading where the neighbours in question are the neighbours of the baby cousin. Discourse (77) also has a reading in which the parents are the parents of the children at the party; this would be represented by the existentially quantified version of the relational noun *parents*. Of course, this would not guarantee that the parents in question are the parents of the party attendees, but it seems like a dubious move to hardwire that knowledge into the semantics for (77). Notice that there is no parallel reading for (78) in which it is the neighbours of the party attendees who looked bored; this is explainable by lack of the existential type-shift for SRNs. There is a second possible reading for (78), as we would expect, on which it is the speaker or hearer's neighbours who looked bored.

In summary, relational nouns can have an externally bound or saturated argument just in case they do not have an oblique argument or occur with an inherent R genitive specifier or complement. The relational argument can be saturated by an existential quantification type-shift for the subclass of weakly relational nouns but not for the subclass of strongly relational nouns. The argument can be bound by a discourse referent, saturated by speaker/hearer indices, or bound by an intrasentential binder.

In what follows we are particularly interested in the contrast in semantic composition between relational nouns and resumptive pronouns. I therefore set aside the possibility of discourse binding. There are two ways in which discourse binding could be handled in this theory. First, we could follow Jacobson (1999, p. 134) in treating discourse-bound and intrasententially bound arguments/pronouns equivalently. I have implicitly adopted this choice here. This means that sentences must be allowed to denote not just type t propositions, but also functions from however many discourse-bound pronouns of type e there are to type t. A sentence containing one discourse-bound pronoun will have type $\langle e, t\rangle$, a sentence with two such pronouns will have type $\langle e, \langle e, t\rangle\rangle$, and so on. As Jacobson (1999) points out, this is not necessarily any more problematic than the denotation of a sentence depending on an assignment function: In both cases, the sentence's denotation is dependent on calculating pronominal reference. Alternatively, discourse-bound arguments/pronouns could be handled by versions of Glue Semantics that use a dynamic meaning language (van Genabith and Crouch 1999; Dalrymple et al. 1999b; Kokkonidis 2003) or linear logic context management (a.k.a. *context threading*; van Genabith and Crouch 1999; Dalrymple 2001).

I similarly set aside saturation by speaker/hearer indices and the existential type-shift, although these additions could be made quite straightforwardly to the static account I present. The basic, simplified generalization is thus that a relational NP is intrasententially bound (or saturated by speaker/hearer indices) unless it has an oblique argument or an inherent R genitive specifier or complement. This is captured by the lexical entry in (79).[6]

[6] Notice that this information does not have to be specified separately in the entry for each relational noun: I assume relevant methods for managing redundancy in the lexicon, such as lexical redundancy rules, lexical templates, or inheritance hierarchies.

(79) *neighbour*: N

$(\uparrow \text{PRED}) = \text{'neighbour'}$

$\lambda y \lambda x.neighbour(x, y) : (\uparrow_\sigma \text{ARG})_e \multimap (\uparrow_\sigma \text{VAR})_e \multimap (\uparrow_\sigma \text{RESTR})_t$

$(\ (\uparrow \text{OBL})_\sigma = (\uparrow_\sigma \text{ARG})\ )$

$$\begin{pmatrix} (\uparrow_\sigma \text{ARG ANT}) = ((\text{GF}^* \ \text{GF} \uparrow) \ \text{GF})_\sigma \\ \lambda z.z \times z : (\uparrow_\sigma \text{ARG ANT})_e \multimap ((\uparrow_\sigma \text{ARG ANT})_e \otimes (\uparrow_\sigma \text{ARG})_e) \end{pmatrix}$$

There are two meaning constructors contributed by the relational noun: the main $\langle e, \langle e, t \rangle \rangle$ type meaning constructor and an optional meaning constructor for binding the relational argument.[7] The main meaning constructor is just like that for non-relational nouns, except that there is an extra type *e* argument. This argument is represented by the s-structure feature ARG(UMENT). This is a motivated move because there is no stable syntactic representation of this argument: it can be realized by an oblique complement, it can be saturated in composition with a specifier or complement genitive NP, or it can be bound externally or saturated by speaker/hearer indices and therefore not be syntactically realized in the relational noun's NP at all.

The realization of the relational argument as an oblique complement is handled by the optional equation that immediately follows the main meaning constructor. I assume that prepositions in oblique arguments to relational nouns make no semantic contribution, a simplifying assumption that is not without precedent (Heim and Kratzer 1998, p. 64). This accounts for examples like (80) and (81), which are the sorts of relational noun phrases discussed in Dalrymple et al. (1999c).

(80) Kim heard a rumour about Sandy.

(81) Mary is the mother of this child.

The equation specifies that the σ-projection of the OBLIQUE grammatical function, *about Sandy* or *of this child*, is the same s-structure resource as the relational noun's argument, ARG. Thus, if the oblique contributes a resource c_σ, then the $(\uparrow_\sigma \text{ARG})$ of the relational noun is c_σ. Resource sensitivity takes care of the rest: the relational noun is

[7] Recall that ANT is an abbreviation of the s-structure feature ANTECEDENT.

looking for a c_σ resource and its oblique argument is contributing one. A quantified oblique argument, such as in (82), works as usual.

(82) Mary is the mother of few children.

The quantified NP *few children* contributes the usual kind of resource for finding its scope: $\forall X.[(c \multimap X) \multimap X]$. In this case it takes the relational noun as its scope.

If the relational noun is in composition with a pre- or postnominal genitive, then the genitive will take the entire $\langle e, \langle e, t \rangle \rangle$ relational noun meaning as its relation argument. It may seem at first that there are potentially bad interactions between this case and the case with an oblique complement that we just looked at. Careful consideration shows that the resource logic in fact handles the various possibilities properly.

Consider first the case where there is a postnominal genitive, realized as the OBLIQUE grammatical function of the relational noun. Recall from (65) that a postnominal genitive NP is assumed to be of type $\langle \langle e, \langle e, t \rangle \rangle, \langle e, t \rangle \rangle$ and that it gets its relational argument from the relational noun:

(65) $\textbf{Kim's}_{post} = \lambda R\ [\lambda x[R(x,\ kim)]]$

Suppose that the relational noun has the following instantiated main meaning constructor (setting aside the optional bound reading meaning constructor for the moment):

(83) $\lambda y \lambda x.neighbour(x,\ y) : a \multimap v_1 \multimap r_1$

In order to take this relational noun as an argument, the postnominal OBLIQUE genitive needs to have an instantiated meaning constructor like the following:

(84) $\lambda R[\lambda x[R(x,\ kim)]] : [a \multimap v_1 \multimap r_1] \multimap (v_2 \multimap r_2)$

In order to access the resources that it needs, the generalized postnominal genitive meaning constructor must refer to $((\text{OBL}\ \uparrow)_\sigma\ \text{ARG})$, $((\text{OBL}\ \uparrow)_\sigma\ \text{VAR})$, and $((\text{OBL}\ \uparrow)_\sigma\ \text{RESTR})$, where a, v_1 and r_1 occur in (84). In particular, the genitive must access the resource that is the ARG of the relational noun. If the optional OBLIQUE equation is absent, then the s-descriptions in the corresponding meaning constructors are sufficient to guarantee that the ARG resources in the relational noun and the genitive are instantiated alike. If the optional OBLIQUE equation is present, then the ARG resource is identified with the resource corresponding to the postnominal genitive itself, but this is completely harmless. As long as the two instances of ARG are the same, the resource logic guarantees that the only correct result is that of the genitive taking the relational noun as an argument. In effect, the name of the ARG resource does not matter in this case, just the identity between ($\uparrow_\sigma$ ARG) in the relational noun's meaning constructor and ((OBL $\uparrow)_\sigma$ ARG) in the genitive NP's meaning constructor and this identity is guaranteed purely by the s-descriptions and the configurational relation that the two NPs occur in. Thus, when the relational noun is in composition with a postnominal genitive NP, it does not matter if the optional OBLIQUE equation is realized or not.

The second case to consider is when the relational noun is in composition with a prenominal genitive and has an oblique argument, as in the following example:

(85) Warhol's portrait of Monroe stinks.

The oblique argument will contribute the normal lexical meaning constructor for a proper name, which is lexically of type e. If the optional oblique equation is not realized and the oblique's resource is not identified as the relational noun's ARG resource, then the derivation will fail due to the resource logic. In particular, if the genitive takes the relational noun's main meaning constructor as an argument, then there will be no consumer for the oblique's resource. Thus, resource sensitivity ensures that when an oblique is present the optional oblique equation must be realized.

This seems to leave us in a bind if the genitive in (85) takes an $\langle e, \langle e, t \rangle \rangle$ argument: the oblique contributes a resource that must be consumed as the relational noun's ARG, the result of which is a type $\langle e, t \rangle$ expression that cannot combine with the genitive. However, notice that the genitive expression in (85) is not an instance of the inherent R genitive: Warhol bears some relation to the portrait, but it is not that of being the depicted entity. The depicted entity is the argument that corresponds to the relational noun's ARG and I just showed that the ARG must be filled by the oblique. In fact, (85) only has a reading in which there is a free, contextually-specified relation between the genitive NP and the noun phrase it is a specifier of. In this case, given the knowledge that Warhol was an artist and that he did indeed do a portrait of Marilyn Monroe, the free relation is likely that of creation; without this knowledge the relation is one of ownership or control. Plugging *Warhol* into Partee's (1983, 1997) semantics for the free genitive (see (67) above), we get:

(86) $\textbf{Warhol's}_{pre/free} = \lambda R_i \lambda P \iota x[P(x) \wedge R_i\,(x, warhol)]$

In particular, the free genitive is looking for an $\langle e, t \rangle$ argument, which is exactly what is provided by a relational noun with an oblique argument. We thus get the following semantics for (85):

(87) $\lambda R_i.stink(\iota x[portrait(x, monroe) \wedge R_i\,(x, warhol)])$

Given Partee's (1983, 1997) theory of genitives, the Glue theory of relational nouns, as exemplified by the lexical entry in (79), correctly predicts that only the free genitive is possible when the oblique argument to the relational noun is present. Notice that this result was achieved due only to the resource sensitivity of linear logic and the lexical specification for relational nouns.

Let us stop to take stock. It has been shown that the lexical entry for relational nouns, exemplified by (79), together with the resource-sensitive Glue logic, linear logic, achieves a number of correct results. It accounts for relational nouns in composition with pre- and post-nominal genitives and relational nouns with non-genitive oblique arguments. The analysis allows prenominal genitives with a relational noun argument to have either a free R or inherent R reading (Stockwell et al. 1973; Partee 1983, 1997). It further correctly predicts, as a consequence of resource sensitivity, that a prenominal genitive has only the free R reading if it is the specifier of a relational noun with an oblique argument. It was also shown that the optional equation that equates the s-structure of the relational noun's ARG to that of its OBLIQUE argument not only allows the relational noun to properly consume its oblique argument's resource, it also interacts properly with pre- and postnominal genitives. The rest of the lexical entry in (79) concerns the bound reading of relational nouns, to which I turn next.

4.1. *The Bound Reading of Relational Nouns*

In this section I will demonstrate how the Glue analysis of relational nouns presented here captures the bound reading. I will also show that the bound reading interacts properly with genitives and obliques. This follows only if the non-optional part of the lexical entry (the main meaning constructor) and the two optional parts do not interfere with each other. We have already seen that the first optional equation interacts properly with the main meaning constructor. It remains to be shown that the optional material that concerns the bound reading interacts properly with the non-optional material and the optional oblique material.

The section of the lexical entry (79) that concerns bound variables has two parts. They are repeated here for ease of reference:

(88) $(\uparrow_\sigma \text{ ARG ANT}) = ((\text{GF}^* \text{ GF} \uparrow) \text{ GF})_\sigma$

(89) $\lambda z.z \times z : (\uparrow_\sigma \text{ ARG ANT})_e \multimap ((\uparrow_\sigma \text{ ARG ANT})_e \otimes (\uparrow_\sigma \text{ ARG})_e)$

The equation (88) is a *binding equation* (Dalrymple 1993, p. 120) of the sort used in LFG's binding theory (Dalrymple 1993, 2001; Bresnan 2001). Binding equations state syntactic constraints on binding and relate bound elements and their binders. Equation (88) identifies the binder (ANT) of the relational noun's argument (ARG). The meaning constructor (89) is a standard type $\langle e, \langle e \times e \rangle \rangle$ pronominal function that takes the binder's resource as an argument to produce the bound argument's resource.

The left side of the binding equation is quite straightforward. It identifies the s-structure node that is found by following the feature path ARGUMENT ANTECEDENT from the s-structure node, $\uparrow_\sigma$, which is the σ-projection of the relational noun's f-structure node, $\uparrow$. Notice that there is a σ subscript that applies to the entire right side of the equation such that the result of resolving the right side is an s-structure node. We are therefore equating two s-structure nodes, the results of the left and right sides.

The right hand side of the equation specifies two things: where the binder of the relational argument may occur and that it bears a grammatical function at f-structure. One side of LFG binding equations, in this case the right one, always has the following general form (Dalrymple 1993, p. 120):

(90) ((DomainPath $\uparrow$) AntecedentPath)

The sub-expression (DomainPath $\uparrow$) is an inside-out functional uncertainty equation (Halvorsen and Kaplan 1988; Dalrymple 1993).

It specifies an f-structure, call it *f*, from which there is a path DomainPath to $\uparrow$. AntecedentPath is the path from *f* to the f-structure of $\uparrow$'s antecedent. AntecedentPath is usually the attribute GF or a more specific instance of GF, such as SUBJ (for example, if the anaphor is subject-oriented). The expression (DomainPath $\uparrow$) is also known as the binding domain (Dalrymple 2001, pp. 283–291). The binding domain is the specification of where the antecedent can occur.

Equation (88) specifies that the binding domain for relational nouns is (GF* GF $\uparrow$). This equation is unpacked as follows. The f-structure variable $\uparrow$ specifies the f-structure node of the relational noun. The equation (GF $\uparrow$) identifies the f-structure node, call it *g*, of the predicate that takes the relational noun as an argument. GF* uses the Kleene star to identify an f-structure node, call it *f*, that is found by moving zero or more GFs out from *g*. The f-structure node identified by (GF* GF $\uparrow$) is either *g*, the f-structure in which the relational noun occurs, or an f-structure that can be found by following a series of GF attributes outward from *g*. Thus, the binding domain (GF* GF $\uparrow$) specifies the possible f-structures within which the relational noun's argument finds the f-structure node that maps to its antecedent at s-structure.

The binding domain of the relational noun, (GF* GF $\uparrow$), is completely unrestricted; i.e., the binding domain is the "Root Domain" (Dalrymple 2001, p. 284). The relational noun finds its antecedent anywhere in this domain. So far we have only discussed the domain in which the antecedent occurs. The actual antecedent is the σ-projection of a grammatical function that occurs within this domain. It is the value of the final GF in (88):

(88) $(\uparrow_\sigma \text{ ARG ANT}) = ((\text{GF}^* \text{ GF} \uparrow) \text{ GF})_\sigma$

The grammatical function that is the antecedent is also unrestricted, then.

Let us see the binding equation (88) at work in the analysis of our bound relational noun sentence, which is repeated here:

(1) Every suburbanite knows a neighbour.

A simplified f-structure for this sentence is shown in (91).

(91)

$$k\begin{bmatrix} \text{PRED} & \text{`know'} \\ \text{SUBJ} & s\begin{bmatrix} \text{PRED} & \text{`suburbanite'} \\ \text{SPEC} & \begin{bmatrix} \text{PRED} & \text{`every'} \end{bmatrix} \end{bmatrix} \\ \text{OBJ} & n\begin{bmatrix} \text{PRED} & \text{`neighbour'} \\ \text{SPEC} & \begin{bmatrix} \text{PRED} & \text{`a'} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Checking the binding domain of *neighbour*'s relational argument, we see that it is satisfied if GF^* is zero and ($\text{GF} \uparrow$), which is ($\text{GF}\ n$) in this case, is the f-structure k. The binding domain is therefore the main f-structure, k. Within this f-structure there is a GF, SUBJ, that can be the relational argument's antecedent.

As things stand, it may appear that the analysis does not ensure that the proper structural relationship holds between the relational noun and its binder. In particular, notice that when the subject and object grammatical functions are swapped in (1), the bound reading disappears and the sentence is only well-formed if the argument is speaker/hearer-saturated or discourse bound:

(92) A neighbour knows every suburbanite.

This sentence does not have a reading in which there is a neighbour X such that for every suburbanite Y, X and Y are neighbours and X knows Y. However, if the f-structures s and n are swapped in (91), we see that the binding domain is still k and there is a GF, OBJ, within this domain. Binding equation (88) thus seems to be satisfied.

This is not the case, though, because LFG's binding theory typically assumes a hierarchical ordering among grammatical functions. Bresnan (2001) uses the Relational Hierarchy (Keenan and Comrie 1977) for syntactic rank:

(93) $\text{SUBJ} > \text{OBJ} > \text{OBJ}_\theta > \text{OBL}_\theta > \text{COMPLEMENT} > \text{ADJUNCT}$

Other versions of binding theory in LFG argue for a thematic hierarchy instead (Dalrymple 1993, pp. 168–177). In either case, a restriction holds that an element lower on the hierarchy cannot bind an element higher on the hierarchy. Therefore, *every suburbanite*, as the OBJ of (92), cannot bind the SUBJ *a neighbour*. Notice that although binding equations relate s-structure nodes, the relational hierarchy is stated over f-structure grammatical functions that map to those nodes. This is similar to the specification of DomainPath and AntecedentPath in terms of f-structures, even though the nodes they restrict will ultimately be the s-structure correlates of the f-structures.

An alternative way to account for this structural asymmetry is to use functional precedence (Bresnan 1984; Kameyama 1985; Kaplan 1989). The idea behind functional precedence (f-precedence) is to use the inverse of the ϕ mapping from constituent structure to functional structure to capture linear precedence relations between f-structures. Intuitively, f f-precedes g if and only if the c-structure material that maps to f linearly precedes the c-structure material that maps to g. Thus, f-precedence boils down to linear precedence in c-structure. Bresnan (1995, 2001) presents an alternative definition of precedence that posits traces in c-structure in tightly circumscribed circumstances; this allows a right-peripheral trace in c-structure to block f-precedence in certain cases and is instrumental in Bresnan's analysis of weak crossover.

There is one piece of evidence that favours either a syntactic rank account in terms of the hierarchy (93) or f-precedence with a trace.[8] The relevant sort of example concerns relational nouns in what is often called 'reconstruction' (Barss 1986; Lebeaux 1988; Chomsky 1993):

(94) How many clients did every lawyer send Christmas cards to?

Unlike example (92), example (94) allows the relational noun *client* to be bound by *every lawyer*, even though the relational noun precedes its binder. The crucial difference between the two examples is that the latter involves a long distance dependency. As in other reconstruction cases, the material in the *wh*-phrase *how many clients* is behaving as if it is at the base of the long distance dependency with respect to binding options. This is in itself significant, since it means that bound relational nouns can serve as a further diagnostic for reconstruction.[9]

[8] Dalrymple et al. (2001) propose an alternative trace-less account of the facts that motivate traces in Bresnan (1995, 2001). At this point it is not clear if their analysis can be extended to these relational noun facts.

[9] It is especially significant because recent evidence has shown that reflexives in English picture NPs – which have often been used as evidence for reconstruction (Chomsky 1993; Fox and Nissenbaum 2004) – do not meet the basic locality requirements that have been postulated for them and are in fact essentially logophoric (Asudeh and Keller 2001; Keller and Asudeh 2001; Runner et al. 2002, 2003).

The bound reading for the relational noun *client* in the reconstruction example (94) is accounted for without movement in LFG, but nevertheless indicates that the requirements of the relational noun's binding equation must be satisfied by *every lawyer*, the subject of the clause where the *wh*-phrase containing the relational noun originates. If f-precedence is interpreted without traces, as in the version of Kaplan (1989), then the binding equation (88) in terms of f-precedence incorrectly blocks the bound reading for (94). If f-precedence is interpreted with a trace in c-structure at the base of the long distance dependency, in the sense of Bresnan (1995, 2001), then it will properly allow the bound reading for (94), as then the rightmost c-structure correspondent of *every lawyer* does precede the trace. Syntactic rank also makes the correct prediction, because the SUBJ *every lawyer* does outrank the grammatical function at the base of the long distance dependency, which is OBJ. The f-structure correspondent of the *wh*-phrase is simultaneously structure shared at f-structure as a discourse function FOCUS and the OBJ. I leave it as a question for further research whether syntactic rank or f-precedence with trace is to be the preferred account of (92) and (94).[10] It suffices for present purposes that the requisite mechanism exists in the theory.

In summary, the binding equation on the relational noun states that the antecedent of the relational noun's argument occurs in the Root Domain (i.e., there are no locality or anti-locality restrictions). Normal assumptions of LFG's binding theory (syntactic rank or f-precedence) ensure that the antecedent and the relational noun are in a proper structural relationship.

4.1.1. *Compositional semantics for bound relational nouns*

We can now turn to an example of the analysis at work in deriving the semantics for (1), our example of a relational noun with a bound reading.

[10] A reviewer points out that data from scrambling languages, such as German, potentially support the syntactic rank approach, since binding can occur with inverted precedence.

(1) Every suburbanite knows a neighbour.

The natural reading of this sentence is one where *every suburbanite* binds *neighbour* such that every suburbanite knows a neighbour of his/her own. This is the *every*-wide reading shown in (95a). The *every*-narrow reading in (95b) is not possible:

(95)a. For every suburbanite X there is a Y such that X and Y are neighbours and X knows Y.

b. There is a Y such that for every sunurbanite X, X and Y are neighbours and X knows Y.

The illicit reading is in effect a crossover violation. We will see below that it is a direct consequence of the logic that only the (95a) reading is possible.

The f-structure for this sentence is repeated here:

(91)
$$k\begin{bmatrix}\text{PRED} & \text{`know'} \\ \text{SUBJ} & s\begin{bmatrix}\text{PRED} & \text{`suburbanite'} \\ \text{SPEC} & \begin{bmatrix}\text{PRED} & \text{`every'}\end{bmatrix}\end{bmatrix} \\ \text{OBJ} & n\begin{bmatrix}\text{PRED} & \text{`neighbour'} \\ \text{SPEC} & \begin{bmatrix}\text{PRED} & \text{`a'}\end{bmatrix}\end{bmatrix}\end{bmatrix}$$

The labels k, n, and s will be used in instantiating the lexically contributed meaning constructors.

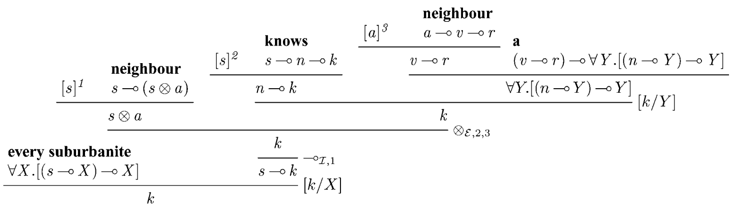
The lexical entries for the determiners and the transitive verb do not bear additional comment, but anticipating the discussion in the next section, the relational noun's entry must be such that only the main meaning constructor and the optional bound argument material is realized. The subject's resource is the ANTECEDENT of the relational argument, as per the discussion in the previous section. I assume the label a for the relational argument's resource and the usual mnemonic labels for the others. I have also taken a shortcut in pre-composing *every* and *suburbanite*. We therefore get the following instantiated meaning constructors:

(96) 1. $\lambda S.every\ (x, suburbanite(x), S(x))$: Lex. **every sub.**
$\forall X.[(s \multimap X) \multimap X]$

2. $\lambda x \lambda y.know(x, y) : s \multimap n \multimap k$ Lex. **knows**
3. $\lambda R \lambda S.a(z, R(z), S(z))$: Lex. **a**
   $(v \multimap r) \multimap \forall Y.[(n \multimap Y) \multimap Y]$
4. $\lambda y \lambda x.neighbour\ (x, y) : a \multimap v \multimap r$ Lex. **neighbour**
5. $\lambda z.\ z \times z : s \multimap (s \otimes a)$ Lex. **neighbour**

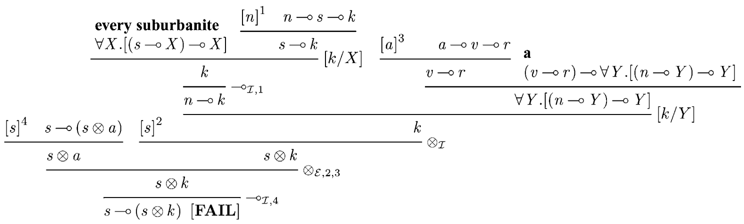
The proof tree in (97) shows the *every*-wide scope reading, the only one possible. Note that I decorate the lexically contributed premises with the corresponding word solely for the reader's convenience; this is not an integral part of the proof.

(97)



The inverse reading, with *a neighbour* taking scope over *every suburbanite*, is precluded by the resource accounting of linear logic. The only way to scope in *a neighbour* is by providing a resource *s* to the relational noun; this must be done by using an assumption, since there is no lexically-contributed resource *s*. This will result in two kinds of resource failure, as shown in the following sketch of the failed proof:[11]

(98)



First, a resource *s* has been assumed and consumed by the relational noun's resource. There are no longer any consumers of *s* left in the derivation, so when the assumption on *s* is discharged there will be an unconsumed resource at the end of the derivation. Second, a separate instance of *s* must be assumed in order to eliminate the $\otimes$ in the relational noun's meaning constructor. But there is no longer a consumer of this *s* resource either. Therefore, there can be no proof

[11] The function *know* : $s \multimap n \multimap k$ is curried to *know* : $n \multimap s \multimap k$.

with *a neighbour* outscoping *every suburbanite*, because there would be two copies of s left and the derivation is thus not a licit Glue derivation. The more general point is that a quantifier cannot scope under an anaphor that is dependent on it. Therefore, the relational argument cannot outscope its binder, *every suburbanite*. A similar point is made by Dalrymple et al. (1999c) with respect to other scopal dependencies as well as oblique arguments of relational nouns.

The following derivation shows the operations in the meaning language for the successful surface scope proof shown in (97). The only tricky step is the use of the operator let. This operator performs simultaneous pairwise substitution (see Section 3).

(99)

1.	$\lambda S.every(x, suburbanite(x), S(x)) :$ $\forall X.[(s \multimap X) \multimap X]$	Lex. **every sub.**
2.	$\lambda x \lambda y.know(x, y) : s \multimap n \multimap k$	Lex. **knows**
3.	$\lambda R \lambda S.a(z, R(z), S(z)) :$ $(v \multimap r) \multimap \forall Y.[(n \multimap Y) \multimap Y]$	Lex. **a**
4.	$\lambda y \lambda x.neighbour(x, y) : a \multimap v \multimap r$	Lex. **neighbour**
5.	$\lambda z.z \times z : s \multimap (s \otimes a)$	Lex. **neighbour**
6.	$u : s$	assumption
7.	$u \times u : s \otimes a$	E$\multimap$, 5, 6, $\Rightarrow_\beta$
8.	$v : s$	assumption
9.	$\lambda y.know(v, y) : n \multimap k$	E$\multimap$, 2, 8, $\Rightarrow_\beta$
10.	$w : a$	assumption
11.	$\lambda x.neighbour(x, w) : v \multimap r$	E$\multimap$, 4, 10, $\Rightarrow_\beta$
12.	$\lambda S.a(z, neighbour(z, w), S(z)) :$ $\forall Y.[(n \multimap Y) \multimap Y]$	E$\multimap$, 3, 11, $\Rightarrow_\beta$
13.	$a(z, neighbour(z, w), know(v, z)) : k$	E$\multimap$, 9, 12, $[k/Y]$, $\Rightarrow_\beta$
14.	let $u \times u$ be $v \times w$ in $a(z, neighbour(z, w), know(v, z)) : k$	E$\otimes$
15.	$a(z, neighbour(z, u), know(u, z)) : k$	14, $\Rightarrow_\beta$
16.	$\lambda u.a(z, neighbour(z, u), know(u, z)) :$ $s \multimap k$	I$\multimap$
17.	$every(x, suburbanite(x),$ $a(z, neighb(z, x), know(x, z))) : k$	E$\multimap$, 1, 16, $[k/X]$, $\Rightarrow_\beta$

The reading is as expected and is glossed as in (95a) above: For every suburbanite X there is a Y such that X and Y are neighbours and X knows Y.

4.1.2. *Interactions with genitives and obliques*

In Sections 2 and 3 we have seen that the analysis of relational nouns accounts for genitives and obliques, the interaction between the two, and the bound reading that is available in their absence. In this section I want to wrap things up with a discussion of how the analysis captures the interaction between the bound reading and genitives and obliques.

The basic generalization is that the bound reading is not available when the relational noun has either an oblique complement or an inherent R genitive specifier or complement:

(100) John knows a neighbour of Mary.

(101) John knows Mary's neighbour.

(102) John knows a neighbour of Mary's.

The bound reading is impossible with a non-genitive oblique, as in (100).[12] The lack of the bound reading for (100) follows from the resource sensitivity of the Glue logic. If the optional anaphoric meaning constructor enters the proof, then a resource corresponding to the relational noun's argument is produced by the anaphoric meaning constructor. But the oblique argument is also contributing a resource. There are then two resources corresponding to the argument, but only one is being consumed by the relational noun's main meaning constructor. This leads to failure in the Glue proof.

The other cases, (101) and (102), could allow a bound reading if the genitive is construed as a free R genitive, but there is insufficient context to allow this. Examples (101) and (102) therefore apparently disallow binding. However, we saw above in the *Warhol* example (86) that a subcase of the free R reading is the agentive reading for relational nouns like *portrait* that allow such a reading. Relational nouns that allow a free R reading with a genitive do allow the bound reading to surface:

(103) Every celebrity was flattered by Warhol's portrait.

This example does have a reading in which every celebrity was flattered by Warhol's portrait of him/her. Implicit in this is a treatment

[12] Recall that examples like (100) were discussed on page 29 above: they are somewhat awkward with strongly relational nouns, but improve if the oblique is a bigger noun phrase.

of the agentive readings of relevant relational nouns as a subclass of free R readings. As an alternative, one might want to specify two arguments at semantic structure for such agentive relational nouns. This would leave unexplained, though, why the agentive argument cannot itself normally be bound:

(104) Everyone saw a portrait of Monroe.

This sentence does not entail that every X saw a portrait of Monroe *by X*. It seems best, then, to treat the agentive reading as a subclass of the free R reading.

Unlike the free R genitive in (103), inherent R genitives do not allow the bound reading, since the inherent R genitive needs to consume the entire type $\langle e, \langle e, t \rangle \rangle$ relational noun meaning. It therefore follows from resource sensitivity that the bound reading is absent when the genitive takes an inherent R, because the only consumer of the implicit argument (the relational noun) has been consumed in its entirety by the inherent R genitive. The resulting contrast between inherent R and free R genitives is shown to be the correct result by the contrast between (101)/(102) and (103). In the latter, the genitive allows a free R reading and a bound reading is available for the relational argument. In (101) and (102), the genitive resists a free R interpretation. The genitive only has an inherent R interpretation and the bound reading is unavailable for the relational argument.

4.1.3 *Interactions with binding theory*

We have thus far seen that the usual mechanisms for binding in LFG can capture the bound reading of relation nouns. However, noun phrases that are headed by common nouns, including relational nouns, cannot in general have antecedents. This is not meant to preclude coreference in the model, but is rather a condition on syntactic binding. The matter seems a little delicate when it comes to relational noun phrases, because the noun phrase as a whole must not have an antecedent, while simultaneously allowing its relational argument at s-structure to have one. This is illustrated by the following example:

(105) *He$_i$ said Mary knows [a neighbour]$_i$.

Notice that the binder of the relational noun does outrank/f-precede it; the binding equation for the relational noun's bound reading therefore does not block this example. Thus, it cannot be that non-relational nouns and relational nouns are distinguished with respect to their ability to take an ANTECEDENT. In this section I will briefly demonstrate that LFG's binding equations and grammatical architecture allow the relational argument to have an antecedent while simultaneously blocking one for the NP as a whole.

There are various ways we could capture the lack of antecedent for non-pronominal noun phrases, but the most straightforward would be a lexical redundancy rule that puts the following constraining equation in the lexical entries of all common nouns:

(106) $\neg\ (\uparrow_{\sigma}\ \text{ANTECEDENT})$

This would get the right result for intrasentential binding, because it would disallow a noun phrase, via its head noun, from having the feature ANTECEDENT at s-structure, no matter its value.

If we are using ANTECEDENT for intersentential binding, too (as in Dalrymple 2001), then we have to be a little more careful, since certain noun phrases, such as definites, can be discourse bound. We would use a negative binding equation, like (107), that states that the ANTECEDENT cannot occur in the Root Domain. Since it is the determiner that decides discourse binding possibilities, we specify the negative binding equation on the determiner, replacing $\uparrow$ with (SPEC $\uparrow$).

(107) $((\text{SPEC}\ \uparrow)_{\sigma}\ \text{ANTECEDENT})\ \neq\ ((\text{GF}^{*}\ \text{GF}\ (\text{SPEC}\ \uparrow))\ \text{GF})_{\sigma}$

This would prevent the noun from taking a commanding ANTECEDENT in the same sentence, but does nothing to preclude discourse binding or coreference in sentences like *After she came in, Mary looked around*. Determiners that block even discourse binding of their noun phrase would simply have a version of the negative binding equation in (106): $\neg\ ((\text{SPEC}\ \uparrow)_{\sigma}\ \text{ANTECEDENT})$. Proper nouns would continue to have a negative binding equation as in (106).

It should be stressed that the negative binding equations can occur on *all* nouns, even relational ones. Recall that the binding equation for relational nouns has as its left side $(\uparrow_{\sigma}\ \text{ARG ANT})$. It is the *argument* of the relational noun that has an antecedent, not the relational noun itself. The equation $(\uparrow_{\sigma}\ \text{ARG ANT})$ does not conflict with the equation $\neg\ (\uparrow_{\sigma}\ \text{ANT})$. Thus, we can simultaneously specify the relational noun as not taking an antecedent while specifying its argument as taking an antecedent without any inconsistency.

This move depends on Glue-LFG's architectural assumption that there is a level of semantic representation, s-structure, which is separate from but related to the syntax. In particular, notice that we could not make the required distinction at the syntactic level, because the relational noun's implicit argument is not even necessarily syntactically realized (i.e., pronounced). Also, since the relational noun is the head of the noun phrase and its f-structural information is therefore identified with the outermost f-structure corresponding to its noun phrase, there is no way to assign an antecedent to the relational noun that is distinct from the antecedent for the whole noun phrase. A similar problem occurs with drawing the distinction between binding the argument versus binding the relational noun itself in the model theory: if the relational noun's binding requirements are passed through in forming the noun phrase, then we lose the distinction between denotations of NPs that can be bound, vs. those that cannot, which was essentially the problem that Jacobson (1999) faced, since for her both pronouns and relational noun phrases are $\langle e, e\rangle$ functions of category NP^{NP}.

4.2. *Summary*

The analysis of relational nouns presented in this section was built around the kind of lexical entry in (108).

(108) $\lambda y \lambda x.neighbour(x, y) : (\uparrow_\sigma \text{ARG})_e \multimap (\uparrow_\sigma \text{VAR})_e \multimap (\uparrow_\sigma \text{RESTR})_t$

$((\uparrow \text{OBL})_\sigma = (\uparrow_\sigma \text{ARG}))$

$$\begin{pmatrix} (\uparrow_\sigma \text{ARG ANT}) = ((\text{GF}^* \ \text{GF} \uparrow) \ \text{GF})_\sigma \\ \lambda z.z \times z : (\uparrow_\sigma \text{ARG ANT})_e \multimap ((\uparrow_\sigma \text{ARG ANT})_e \otimes (\uparrow_\sigma \text{ARG})_e) \end{pmatrix}$$

The analysis of relational nouns captures their interactions with pre- and postnominal genitives and oblique arguments. It also makes the proper prediction that a prenominal genitive specifier of a relational noun with an oblique argument can only have the free reading. The bound reading of relational nouns is captured by the second, optional meaning constructor and the binding equation. Resource sensitivity ensures that the bound reading cannot occur with an oblique argument. It was demonstrated that the binding mechanism can block the relational noun phrase as a whole from taking an antecedent, while allowing the argument at s-structure to take one. The analysis also captures properties of relational nouns with respect to crossover and reconstruction. It was shown that the bound reading properly prevents a crossover violation of the relational noun outscoping its binder, purely based on the resource logic. Similarly, it was shown that existing LFG approaches to binding prevent the relational noun from preceding its binder in just those places where it is appropriate.

5. Resumptive Pronouns

Any theory of resumptive pronouns must capture the following key generalization: the resumptive pronouns of a language L are always the regular pronominal forms of L (McCloskey 2002, p. 192). No language has been described that clearly has a special paradigm for resumptive pronouns. This means that the lexical specifications for resumptive pronouns should not differ from the specifications of non-resumptive pronouns, because then the absence of distinguishing morphology would be unexplained. As a consequence, it is not the pronouns themselves that license resumption, but rather some other element in the pronouns' environment. The nature of the resumptive licenser has been the central focus of research in resumptive pronouns, particularly in the literature on Irish and Welsh, since in these languages the licenser is a morphologically distinguished particle (McCloskey 1979, 1990, 2002; Sells 1984; Willis 2000). The particle is best analyzed as a complementizer (McCloskey 1979, 1990, 2002). This points the way to a general theory of resumptive pronoun licensing that ties resumptive pronouns to lexical properties of complementizers. The licensing complementizer is either in the clause containing the resumptive or in a higher clause. The relationship between the complementizer and the resumptive pronoun is unbounded and acyclic (McCloskey 1979). This follows from the fact that the resumptive pronoun is a regular pronoun and the fact that anaphoric relations are in general unbounded (subject to the locality effects of binding theory) and acyclic.

In the theory of resumption presented in Asudeh (2004), which I review here, the resumptive licenser is a meaning constructor called a *manager resource*. Manager resources are lexically contributed by a resumptive-licensing complementizer and have the following general compositional schema, where P is some pronoun in the material that the complementizer introduces and A is the antecedent or binder of P:

(109) $(A \multimap A \otimes P) \multimap (A \multimap A)$

Notice that the antecedent of the main implication in (109) has the form of a pronominal meaning constructor, which means that a manager resource needs to consume a pronominal resource. The result of this consumption is an implicational modifier resource on the binder. The resources corresponding to the manager resource, the resumptive pronoun and the binder of the resumptive pronoun together yield just the binder:

(110)

1.	A	Lex. (**antecedent**)
2.	$A \multimap (A \otimes P)$	Lex. (**pronoun**)
3.	$[A \multimap (A \otimes P)] \multimap (A \multimap A)$	Lex. (**manager resource**)
4.	$A \multimap A$	E $\multimap$, 2, 3
5.	A	E $\multimap$, 1, 4

It is important that the consequent of the main implication in the manager resource is itself an implication on the pronoun's binder ($A \multimap A$), rather than just another instance of the binder's resource (A). In the latter case, there would be a new copy of the resource A and this would lead to a resource management problem, as there would be two copies of A where only one is required.

The basic function of the manager resource is to remove the pronoun from composition. Thus, a resumptive pronoun that is licensed by a manager resource behaves syntactically exactly like a non-resumptive pronoun, but behaves semantically like a gap: the semantic argument position corresponding to the pronoun gets bound by the pronoun's binder rather than being saturated by the pronoun. The fact that a manager resource removes a pronoun from semantic composition is reflected in the meaning side of the manager resource's meaning constructor by the vacuous lambda abstraction over the pronoun's function. The manager resource consumes the pronoun's meaning, letting the rest of the semantic derivation proceed as if the pronoun had been absent. We can see this by considering proof (110) with the meaning side of the meaning constructors added:

(111)

1.	$a : A$	Lex. (**antecedent**)
2.	$\lambda z.z \times z : A \multimap (A \otimes P)$	Lex. (**pronoun**)
3.	$\lambda P \lambda y.y :$ $[A \multimap (A \otimes P)] \multimap (A \multimap A)$	Lex. (**manager resource**)

4. $(\lambda P \lambda y.y)(\ \lambda z.\ z \times z) : A \multimap A$ E $\multimap$, 2, 3
5. $\lambda y.y : A \multimap A$ 4,$\Rightarrow_\beta$
6. $(\lambda y.y)(a) : A$ E$\multimap$, 1, 5
7. $a : A$ 6,$\Rightarrow_\beta$

It is worth reiterating that the effect of a manager resource is to remove a pronoun from semantic composition but that in the *syntax* there is no difference between resumptive and non-resumptive pronouns.

At this stage it will be useful to look at an example to see in some detail how the syntax and semantics of resumptives work according to this theory. I will again abstract away from language-particular details by using English words for expository purposes. This should *not* be taken as an implicit claim that English has resumptive pronouns.

(112) Every clown WH_{pro} Mary knows him laughed.

Again, let us suppose that WH_{pro} is a resumptive-sensitive complementizer.

Simplifying somewhat, we get the following meaning constructors from the lexical items:

(113)
1. $\lambda R \lambda S.every(x, R(x), S(x)) :$ $(v \multimap r) \multimap \forall X.[(c \multimap X) \multimap X]$ Lex. **every**
2. $clown : v \multimap r$ Lex. **clown**
3. $\lambda P \lambda Q \lambda z.Q(z) \wedge P(z) :$ $(p \multimap k) \multimap [(v \multimap r) \multimap (v \multimap r)]$ Lex. $\mathbf{WH}_{\text{pro}}$
4. $\lambda P \lambda x.x : [c \multimap (c \otimes p)] \multimap (c \multimap c)$ Lex. $\mathbf{WH}_{\text{pro}}$
5. $mary : m$ Lex. **Mary**
6. $\lambda x \lambda y.know(x, y) : m \multimap p \multimap k$ Lex. **knows**
7. $\lambda z.z \times z : c \multimap (c \otimes p)$ Lex. **him**
8. $laugh : c \multimap l$ Lex. **laughed**

Note in particular that the relative complementizer WH_{pro} is contributing two meaning constructors. The first is the normal meaning constructor for a restrictive relative clause, a modifier on the relativized noun's meaning (for a more detailed exposition of restrictive relatives in Glue, see Dalrymple 2001, pp. 400–426). The second meaning constructor is the manager resource.

The following proof shows how the lexically-contributed linear logic resources in (113) compose the meaning of the sentence (the operations in the meaning language follow from the Curry–Howard isomorphism, but are also shown in detail in (115) below). The proof is broken into three parts, solely for presentational purposes. The first part, (114a), shows how the manager resource disposes of the pronominal resource and how the resulting resource for the pronominal binder is threaded through the resource for *laugh*, the predicate that takes it as an argument. The second part, (114b) shows the modificational step of composing the relative clause with the head it modifies. The conclusion of the proof fragment (114b) serves as the restriction to the quantifier *every* and the conclusion of the proof fragment (114a) serves as the nuclear scope of the quantified NP *every clown*; this is indicated via the boxed references to (114a) and (114b). It is important to bear in mind that this is actually one proof, though, whose presentation is limited by the width of the page.

(114) (a)

$$\dfrac{\dfrac{\dfrac{\dfrac{\stackrel{\textbf{WH}_{pro}\ \textbf{(manager)}}{[c \multimap (c \otimes p)] \multimap (c \multimap c)} \qquad \stackrel{\textbf{him}}{c \multimap (c \otimes p)}}{(c \multimap c)} \qquad [c]^1}{c} \qquad \stackrel{\textbf{laughed}}{c \multimap l}}{\dfrac{l}{c \multimap l}\ {\multimap_{\mathcal{I},1}}}}{}$$

(b)

$$\dfrac{\dfrac{\dfrac{\stackrel{\textbf{Mary}}{m} \qquad \stackrel{\textbf{knows}}{m \multimap p \multimap k}}{p \multimap k} \qquad \stackrel{\textbf{WH}_{pro}}{(p \multimap k) \multimap [(v \multimap r) \multimap (v \multimap r)]}}{(v \multimap r) \multimap (v \multimap r)} \qquad \stackrel{\textbf{clown}}{(v \multimap r)}}{(v \multimap r)}$$

(c)

$$\dfrac{\dfrac{\boxed{\textbf{114b}} \qquad \stackrel{\textbf{every}}{(v \multimap r) \multimap \forall X.[(c \multimap X) \multimap X]}}{\forall X.[(c \multimap X) \multimap X]} \qquad \boxed{\textbf{114a}}}{l}\ [l/X]$$

As discussed above, the manager resource removes the pronoun from composition (the first line of (114a)), clearing the way for the argument corresponding to the pronoun in the semantics to be bound by the pronominal binder, *every clown*, just as if the relative clause had been a non-resumptive relative.

The operations on the meaning terms follow by the Curry–Howard isomorphism, but the details are also shown in the list derivation (115):

(115)

1.	$\lambda R\lambda S.every(x, R(x), S(x))$: $(v \multimap r) \multimap \forall X.[(c \multimap X) \multimap X]$	Lex. **every**
2.	$clown : v \multimap r$	Lex. **clown**
3.	$\lambda P\lambda Q\lambda z.Q(z) \wedge P(z)$: $(p \multimap k) \multimap [(v \multimap r) \multimap (v \multimap r)]$	Lex. $\textbf{WH}_{pro}$
4.	$\lambda P\lambda x.x : [c \multimap (c \otimes p)] \multimap (c \multimap c)$	Lex. $\textbf{WH}_{pro}$
5.	$mary : m$	Lex. **Mary**
6.	$\lambda x\lambda y.know(x, y) : m \multimap p \multimap k$	Lex. **knows**
7.	$\lambda z.z \times z : c \multimap (c \otimes p)$	Lex. **him**
8.	$laugh : c \multimap l$	Lex. **laughed**
9.	$\lambda x.x : (c \multimap c)$	E $\multimap$, 4, 7, $\Rightarrow_\beta$
10.	$u : c$	assumption
11.	$u : c$	E $\multimap$, 9, 10, $\Rightarrow_\beta$
12.	$laugh(u) : l$	E $\multimap$, 8, 11
13.	$\lambda u.laugh(u) : c \multimap l$	I $\multimap$
14.	$\lambda y.know(mary, y) : p \multimap k$	E $\multimap$, 5, 6, $\Rightarrow_\beta$
15.	$\lambda Q\lambda z.Q(z) \wedge know(mary, z)$: $(v \multimap r) \multimap (v \multimap r)$	E $\multimap$, 3, 14, $\Rightarrow_\beta$
16.	$\lambda z.clown(z) \wedge know(mary, z) : v \multimap r$	E $\multimap$, 2, 15, $\Rightarrow_\beta$
17.	$\lambda S.every(x, clown(x)$ $\wedge\ know(mary, x), S(x))$: $\forall X.[(c \multimap X) \multimap X]$	E $\multimap$, 1, 16, $\Rightarrow_\beta$
18.	$every(x, clown(x)$ $\wedge\ know(mary, x), laugh(x)) : l$	E $\multimap$, 12, 17, $[l/X]$, $\Rightarrow_\beta$

In sum, the key ideas of this theory of resumptive pronouns are the following:

1. Resumptive pronouns are not distinguished in any way from non-resumptive pronouns.
2. A resumptive reading of a pronoun occurs only in the presence of a suitable manager resource.
3. Manager resources remove resumptive pronouns from semantic composition.
4. Manager resources act at the syntax–semantics interface: the pronouns are present and undistinguished in the syntax.

We will see in the next section that this theory solves the resumptive puzzle because the manager resources that license resumptive pronouns fail to license relational nouns. The key to making the distinction is in the s-description of the manager resource, the most general form of which is:

(116) $[((\uparrow \text{GF}^{+})_{\sigma}\ \text{ANT})_{e} \multimap [((\uparrow \text{GF}^{+})_{\sigma}\ \text{ANT})_{e} \otimes (\uparrow \text{GF}^{+})_{\sigma_e}]]$
$\multimap [((\uparrow \text{GF}^{+})_{\sigma}\ \text{ANT})_{e} \multimap ((\uparrow \text{GF}^{+})_{\sigma}\ \text{ANT})_{e}]$

This s-description has two constituent equations, $((\uparrow \text{GF}^{+})_{\sigma}\ \text{ANT})$ and $(\uparrow \text{GF}^{+})_{\sigma}$. The feature ANT(ECEDENT) is proper to semantic structures and therefore does not need to be σ-mapped. The feature GF is short for any f-structural grammatical function and the specification $(\uparrow \text{GF}^{+})_{\sigma}$ uses Kleene plus to indicate that it can be satisfied by the σ-projection of a grammatical function in the f-structure of the manager resource's contributor (designated by $\uparrow$) or by an arbitrarily deeply-embedded grammatical function. Notice that nothing guarantees that the instances of $(\uparrow \text{GF}^{+})_{\sigma}$ get instantiated to the same s-structure node. However, the resource sensitivity of linear logic guarantees that a successful proof will be found only if this is indeed the case, in which case we get the schematic form of the manager resource familiar from (109) above:

(109) $(A \multimap A \otimes P) \multimap (A \multimap A)$

Notice that the linear logic atoms are in fact *typed* (see section 3 above and appendix A.1). A manager resource is therefore of type $\langle\langle e, \langle e \times e\rangle\rangle, \langle e, e\rangle\rangle$.[13]

[13] There is another way to exercise more control over the realization of separate instances of $(\uparrow \text{GF}^{+})_{\sigma}$, which might be desirable in computational applications, since the prover would be prevented from attempting certain proofs that are known to fail. The method uses *local names* (Kaplan and Maxwell 1996), which are f-structure variables that have scope only in the lexical item or rule element in which they occur (Dalrymple 2001, pp. 146–148). Using a local name %RP, the manager resource in (116) above is specified as follows (Asudeh 2004, p. 153):

(i) $\%\text{RP} = (\uparrow \text{GF}^{+})$
$[(\%\text{RP}_{\sigma}\ \text{ANT}) \multimap ((\%\text{RP}_{\sigma}\ \text{ANT}) \otimes \%\text{RP}_{\sigma})]$
$\multimap [(\%\text{RP}_{\sigma}\ \text{ANT}) \multimap (\%\text{RP}_{\sigma}\ \text{ANT})]$

The local name %RP is set to the f-structure of the resumptive pronoun. Every instance of %RP in the scope of the lexical item that contributes the manager resource refers to this same f-structure node.

5.1. *Summary*

The basic idea behind this theory of resumptive pronouns is that the problem of resumption is a problem of resource surplus: the resumptive pronoun's resource apparently goes unconsumed. The consumer of the resource is a manager resource and it is the presence of a manager resource that licences a resumptive use of a pronoun. Manager resources are lexically specified and operate at the syntax–semantics interface. The result is a theory of resumptives that treats resumptive pronouns as ordinary pronouns in the syntax and ties their exceptional ability to occur at the base of a long distance dependency to the presence of a manager resource. This has been a necessarily brief overview of Asudeh's (2004) theory of resumptive pronouns in Glue Semantics. Enough is in place, though, to demonstrate how Glue solves the resumptive puzzle of relational nouns.

6. A Solution to the Resumptive Puzzle and its Consequences

At the beginning of this paper, I presented the following puzzle about relational nouns, which I named the *resumptive puzzle*:

(20) If the implicit argument of a relational noun can be bound like a pronoun, why is it impossible for the argument to function resumptively?

I showed that this puzzle is a genuine problem for semantic theory and that both the variable-free theory of Jacobson (1999) and the variable-based theory of Heim and Kratzer (1998) failed to solve it.

On the variable-free theory developed here, both pronouns and relational nouns (on the bound reading) are functions on their antecedents, just as in Jacobson's theory. Pronouns and relational nouns have the same syntactic category, which maintains the SWUP generalization that nouns and pronouns have the same distribution. Lastly, the same binding mechanism accounts for the binding of pronouns and relational nouns. However, the architecture of Glue Semantics allows resumptive licensers (manager resources) to distinguish between the two. The key insight is the following:

(117) Pronouns are syntactic arguments, but relational arguments are not – they are purely semantic arguments.

It follows that pronouns are visible to syntactic processes, such as resumptive licensing, whereas relational arguments are not. The architecture of the theory allows us to capture this distinction: Pronouns are present in the syntactic levels of c-structure and/or f-structure (null pronominals are present only in f-structure), but relational arguments are not present in syntactic structure; they are only present in semantic structure.

This solution to the resumptive puzzle formalizes Partee's (1989) claim that relational arguments are semantically present but syntactically absent. The resumptive puzzle in fact applies more generally than to just relational nouns. Any covert variables introduced into the syntax, for example as proposed in recent work by Stanley (2000) and Martí (2001, 2003, 2004), would be expected to function resumptively, all else being equal. Although at this point it is an open empirical question, it seems unlikely that the syntactic variables in question could be resumptives. It is possible that independent factors might bar resumptive uses of some of these variables, but a potentially promising general solution would be to extend the present solution to these other cases. However, this would entail that the variables in question are in fact aspects of semantic representation, as argued by Partee (1989), and not present in the syntax after all. Stanley's and Martí's proposals would then have to be reconsidered.

The present theory formally captures (117) as follows. Recall that the generalized form of manager resources is as follows:

(118) $$[((\uparrow \text{GF}^{+})_{\sigma}\ \text{ANT})_{e} \multimap [((\uparrow \text{GF}^{+})_{\sigma}\ \text{ANT})_{e} \otimes (\uparrow \text{GF}^{+})_{\sigma_{e}}]] \multimap [((\uparrow \text{GF}^{+})_{\sigma}\ \text{ANT})_{e} \multimap ((\uparrow \text{GF}^{+})_{\sigma}\ \text{ANT})_{e}]$$

A manager resource is looking for some grammatical function that has an antecedent. But relational nouns do not have antecedents: their *arguments* at s-structure have antecedents. Therefore, even though relational nouns on the bound reading contribute a pronominal meaning constructor, the equation $((\uparrow \text{GF}^{+})_{\sigma}\ \text{ANT})$ cannot be satisfied by the relational noun. Only an equation of the form $((\uparrow \text{GF}^{+})_{\sigma}\ \text{ARG ANT})$ could pick up the correct resource from the relational noun, but this is not the kind of equation that a manager resource is constructed with. Furthermore, the equation $((\uparrow \text{GF}^{+})_{\sigma}\ \text{ARG ANT})$ could not pick up a pronominal resource, since pronouns do not have internal arguments. Therefore, the theory strictly separates pronouns from relational arguments.

The analyses of relational nouns and resumptive pronouns correctly account for resumptive pronouns within relational NPs, as in the following Irish example:

(119)	an	fear	a	bhfuil	**a**	mháthair	san	otharlann
	the	man	COMP	is	his	mother	in.the	hospital

the man who his mother is in the hospital
(McCloskey 1979, 6, (4))

This example contrasts with the impossible cases of binding a relational argument, because in this case there is an actual resumptive pronominal possessor of the relational noun. The resumptive pronoun *a* ('his') is consumed by a manager resource that is contributed by the resumptive-sensitive complementizer *a*. The relative head *an fear* ('the man') is interpreted in the position of the resumptive. The result is that (119) is interpreted equivalently to *the man whose mother is in the hospital*. The possessor of the relational noun *mháthair* ('mother') is most readily interpreted as an inherent prenominal genitive: The *mother* relation contributed by the noun saturates the prenominal genitive's $\langle e, \langle e, t \rangle \rangle$ relation argument. As discussed in Section 4.1.2, if the prenominal genitive can be construed as a free genitive, then the bound reading of the relational noun occurs, with the relational noun's implicit argument potentially bound by the relative head *an fear*. However, the relational argument would not be resumptive on *an fear*. It would simply be bound in the usual manner.

7. Glue and Direct Compositionality

The analyses of relational nouns and resumptive pronouns yield a solution to the resumptive puzzle that rests in part on the grammatical architecture of Glue Semantics. It seems, though, that crucial use has been made of a level of representation (semantic structure) in a way that violates the hypothesis of direct compositionality (Jacobson 1999, 2002) – the hypothesis that surface structures receive a direct model-theoretic interpretation. Jacobson (2002) considers four theories of the syntax-semantics interface. The first three, she argues, are increasingly weaker versions of direct compositionality (*strong direct compositionality*, *weak(er) direct compositionality*, and *deep compositionality*). The difference between these boils down to their increasingly elaborated views of syntactic structure. In particular, strong direct compositionality assumes only a context-free phrase structure grammar or equivalent (Jacobson 2002, p.603); syntactic structure is built through string concatenation. Jacobson (2002) argues that these varieties of direct compositionality contrast as a group with the fourth theory, which postulates a level of Logical Form (LF) to which surface structures are mapped and which then receives a model-theoretic interpretation.

This section has two main goals. The first is to show that appearances are deceptive: Glue Semantics as presented here is in fact directly compositional and strongly so. The second goal is explore the implications of Glue Semantics for the hypothesis of direct compositionality and for variable-free semantics and to discuss avenues for future research.

7.1. *Directly Compositional Glue*

The analyses presented here have been couched in Glue Semantics for Lexical Functional Grammar. The architecture of LFG is parallel and modular. Different levels of grammatical representation are described in parallel using structures and logics that are appropriate to the level in question. Distinct data structures encode different kinds of representations: constituent structure is represented with trees, functional structure is represented by sets of attribute-value pairs which are equivalent to tabular functions (Kaplan 1987, p. 352). The separate representations are related by structural correspondences (a.k.a. projection functions), such as the ϕ projection from constituent structure to functional structure and the σ projection from functional structure to semantic structure (Kaplan 1987, 1989). This was shown in (34) in Section 3, which I repeat here:

(34) constituent structure $\xrightarrow{\phi}$ functional structure $\xrightarrow{\sigma}$ semantic structure

But this is just part of the projection architecture that Kaplan (1987, 1989) discusses. In particular, there is a mapping to c-structure from the string:

(120) string $\xrightarrow{\pi}$ c-structure $\xrightarrow{\phi}$ f-structure $\xrightarrow{\sigma}$ s-structure

Glue Semantics based on this architecture can be shown to satisfy direct compositionality.

Notice first that the correspondences can be composed, since the domain of each successive function is the range of the previous one. Given this, the following quote from Kaplan (1987, p. 363) is quite relevant to the discussion of direct compositionality.

> Although the structures related by multiple correspondences might be descriptively or linguistically motivated levels of representation, justified by sound theoretical argumentation, they are formally and mathematically, and also computationally, eliminable ... Obviously there is a structural correspondence that goes from the word string to the f-structure, namely the composition of π with ϕ.

There are two key points in this passage. First, intermediate levels are eliminable through composition of correspondence functions. Second, although such elimination is possible, it may nevertheless be desirable to have separate levels. I will pick up on both of these points in what follows.

Kaplan observes that we can compose π and ϕ to go directly from strings to f-structures. We can further compose $\pi \circ \phi$ with σ, moving directly from the string to semantic structure, the resources of the Glue logic. The nature of these mapping functions is important to consider. First, the mapping π from strings to c-structures is to be understood as a family of functions such that for each string there is a set of π functions mapping the string to different structural analyses (c-structures). The set of π functions may be the null set (for a string with no c-structure parse), a singleton (for a parseable but unambiguous string), or else contains as many different instances of π as required for multiple c-structure parses of an ambiguous string. Similarly, a string may have only one c-structure, but there may be multiple instances of the ϕ mapping if the c-structure is f-structurally ambiguous. The same comments apply to the σ mapping from f-structure to s-structure.

Second, since lexical entries simultaneously contain information about each item's c-structure, f-structure, and s-structure – respectively, in the form of categorial information, functional equations, and meaning constructors – there is a mapping from words to s-structure elements, which are linear logic resources. A linear logic proof is then conducted on these resources. This proof is equivalent to Categorial Grammar proofs in providing semantic composition, i.e. the syntax of the semantics. Linear logic is commutative, whereas Categorial Grammar is in general not. CG thus brings together the syntax of parsing and the syntax of composition. In the version of Glue Semantics that I have been sketching in this section, the syntax is in the composed correspondence $\pi \circ \phi \circ \sigma$. The linear logic proof is thus the pure syntax of composition. The difference between Glue and Categorial Grammar is that in the latter the syntax of the semantics is the *only* syntax, whereas in the former there is a level of syntax beyond the syntax of composition. These are two different architectural designs and each has its merits. Allowing for a syntax of composition, which all semantic theories must allow for, I have just shown that we can define a version of Glue Semantics in the LFG architecture that satisfies Jacobson's (2002) strong direct compositionality: surface structure strings are directly interpreted compositionally.

The second point that Kaplan makes is that although the levels are eliminable, it may nevertheless be useful to work with separate levels. I showed that the solution to the resumptive puzzle depended on distinguishing at semantic structure the antecedent of a pronoun from the antecedent of an argument of a relational noun; namely ($\uparrow_\sigma$ ANT) vs. ($\uparrow_\sigma$ ARG ANT). The manager resources that license resumptive pronouns are sensitive to this distinction. The 'compiled' view of the grammar with $\pi \circ \phi \circ \sigma$ obscures this difference between pronouns and relational nouns, but it is still there. Relational nouns and pronouns are both of the syntactic category N and therefore have the same distribution at constituent structure. On the function composition view sketched here, the function from a pronoun to its model theoretic semantics is the same as the function for a relational noun's bound reading *up to s-structure*. The pronoun finds its antecedent by identifying an s-structure node that satisfies ($\uparrow_\sigma$ ANT). On the bound reading, the relational noun must identify an s-structure node that satisfies ($\uparrow_\sigma$ ARG ANT). It is true that the s-descriptions are distinct, but on the compiled view of the grammar we do not have access to them. All we have access to are the *nodes* in s-structure that these descriptions describe. It is these nodes that are the resources for the linear logic proofs. The manager resource's s-description will be resolved in such a way that only a proof that contains a pronoun can satisfy the s-description, but on this view of the grammar the reason is not obvious: All we see is that when we plug in a pronoun the resources named by the pronoun match up correctly with the resources named by the manager resource and that when we plug in a relational noun the relevant resources do not match up correctly. The reason *why* – the fact that pronouns take an antecedent directly but relational nouns have an internal argument that takes an antecedent – is hidden. The compiled grammar is directly compositional, since it involves no rearrangement of syntactic structures, and it gets the correct results, but it is linguistically less illuminating.

The resumptive puzzle could equally be addressed in other variable-free approaches, such as Jacobson's, or in variable-based approaches like that of Heim and Kratzer (1998). However, the solution presented here rests on architectural distinctions that are not made in these theories, in particular the presence of a linguistic level of semantic representation (s-structure). Therefore this solution is not readily available to these other theories. A similar solution could be attempted in a Heim/Kratzer-style theory by distinguishing the kind of *pro* argument found in relational nouns from null and overt pronominals. Care would have to be taken, though, that the binding mechanism for ordinary pronouns can still bind relational arguments. Similarly, a solution in a Jacobson-style variable-free semantics might be attempted by assigning relational nouns a different category from pronouns or by adding some distinguishing feature. However, the result would potentially endanger the SWUP generalization. The type shifts that maintain SWUP for pronouns would have to be extended to relational nouns, at the expense of complicating the overall system and arguably losing a strong notion of the generalization.

7.2. *Implications for Direct Compositionality and Variable-Free Semantics*

The grammatical architecture of Glue-LFG gives a useful perspective on direct compositionality, which is an appealingly restrictive theoretical position. The Glue perspective is that on the one hand grammatical architectures should have a design that makes direct compositionality in principle available, but on the other hand architectures should also provide a facility for taking the mapping from surface structures to models apart, because otherwise-hidden linguistically relevant generalizations may emerge. Indeed, this sort of architecture is arguably what Montague's PTQ (Montague 1974), with its intermediate representation of Intensional Logic, yields. The Intensional Logic may provide a useful level for stating generalizations. We just have to be careful that the theoretical machinery behind the generalizations can be realized directly. This may lose the generalization, but it shows that the generalization has not made crucial use of a contingent level of representation.

Some recent work in Glue Semantics has explicitly denied direct compositionality, though. This work has investigated the possibility of using properties of the linear logic proofs themselves to explain linguistic phenomena, such as grammatical violations of the Coordinate Structure Constraint (Asudeh and Crouch 2002a) and scope parallelism in ellipsis (Asudeh and Crouch 2002b). This is possible because the linear logic proofs have strong identity criteria and can therefore be reasoned about and manipulated without risk of being misled by contingent facts of representation: There is no danger of mistaking mere pictures for a semantic theory.[14] This work in part constitutes an extended argument that proof theory, as well as model theory, has linguistic relevance. This is clearly a true denial of direct compositionality, because the proofs cannot be compiled away. The basic insight behind this line of Glue research is that the results and methods of proof theory have serious linguistic potential and that linguistic semantics should make room for proof theory alongside model theory.

This paper has demonstrated that a version of Glue Semantics that is compatible with direct compositionality is also possible and that it can yield analyses of puzzling empirical phenomena, such as the lack of resumptive binding of relational nouns. There is no contradiction in these two lines of research. At this point all it means is that some Glue analyses are directly compositional and some are not. If the proof-theoretic line of research proves convincing in the long run, then it calls the hypothesis of direct compositionality into question. This does not mean that previously established directly compositional analyses, in Glue or not, are invalidated. They could still be good analyses of the phenomena in question. It is not the case that every analysis must make use of proof theory even if there is a linguistically relevant level of proofs between surface structure and the model theory. It also does not mean that Logical Form is necessarily exculpated. It would first have to be demonstrated that LF has sound proof-theoretic properties.

The two strands of Glue theory also yield a different perspective on the variable-free theory of semantics. Jacobson (1999) argues that variable-free semantics follows naturally from the hypothesis of direct compositionality. However, even versions of Glue Semantics that deny direct compositionality do not deny the variable-free analysis of anaphora. Thus, even if one rejects direct compositionality, variable-free

[14] I owe this way of putting things to Dick Crouch (p.c.).

semantics may be appealing on other grounds, such as its elimination of assignment functions and the resulting simpler treatment of pronouns.

8. Conclusion

I have presented an analysis of relational nouns that treats their relational arguments as semantic arguments that are absent from the syntax. This allowed bound readings of relational nouns while blocking relational noun phrases as a whole from taking antecedents. Bound relational arguments were investigated with respect to scope, reconstruction, and crossover. I also showed that the bound reading interacts properly with overt genitive and oblique realizations of the relational argument.

The paper began by considering the resumptive puzzle about relational nouns: If the implicit argument of a relational noun can be bound like a pronoun, why is it impossible for the argument to function resumptively? I presented a theory of resumptive pronouns (Asudeh 2004) and showed that the theory of resumption together with the theory of relational nouns solves the resumptive puzzle. The manager resources that license resumptive pronouns consume anaphoric resources that are syntactic arguments. Bound relational nouns contribute anaphoric resources, but these are semantic arguments, not syntactic arguments. Therefore, implicit arguments of relational nouns cannot be licensed as resumptives. The solution to the resumptive puzzle formally captures Partee's (1989) claim that relational arguments occur in the semantics, but not in the syntax. I noted that this has potentially wider consequences for recent work that postulates a wide variety of syntactic variables, which equally raise the resumptive puzzle.

The solution to the resumptive puzzle relied on aspects of Glue Semantics and the architecture of Lexical Functional Grammar, in particular the presence of a separate level of semantic structure. The paper ended by considering the implications of this for the hypothesis of direct compositionality. It was demonstrated using the LFG architecture that the solution can satisfy strong direct compositionality by mapping directly from the strings to the models. However, I argued that this masks the linguistic generalization behind the solution – that relational nouns and pronouns are distinguished at s-structure in terms of how they take antecedents. This supports Kaplan's (1987) claim that formally eliminable levels of representation may nevertheless be linguistically useful and leads to an understanding of direct compositionality in which the hypothesis is satisfied so long as intermediate levels are *in principle* eliminable. I also considered a strand of work in Glue Semantics that directly denies direct compositionality by making crucial use of the linear logic proofs to state linguistic constraints. This work is appealing because it opens up a little-travelled path in semantic research: the possibility of proof theory taking its place alongside model theory in linguistic semantics. However, I emphasized that at this point the two varieties of Glue analysis co-exist, which promises to yield future benefits in terms of a subtle understanding of direct compositionality and the possibility of further integrating proof theory into semantics.

Appendix A: Glue Using Mill

In this Appendix, I define the Glue logic in terms of the indicated fragment of linear logic. In Section A.1 I define the meaning language, the fragment of linear logic, and the Glue logic that puts them together. The presentation follows Dalrymple et al. (1999a,b) and especially Crouch and van Genabith (2000). In Section A.2 I present Prawitz-style natural deduction proof rules for the multiplicative (⊗), modality-free (no ! or ? modalities) fragment of intuitionistic linear logic (MILL), following presentations by Crouch and van Genabith (2000), Benton et al. (1993), Troelstra (1992), Girard (1995), and Dalrymple et al. (1999a). In Section A.3 I give the Curry–Howard term assignments for the meaning language, following the presentations of Glue meaning language term assignment by Dalrymple et al. (1999a) and Crouch and van Genabith (2000) and general presentations of Curry–Howard term assignments by Abramsky (1993), Benton et al. (1993), and Gallier (1995).

A.1. *The Glue Logic*

$$
\begin{array}{llll}
\langle\text{meaning}\rangle & ::= & \langle\text{meaning-const}\rangle & \text{(constants)} \\
 & \mid & \langle\text{meaning-var}\rangle & \text{(variables)} \\
 & \mid & \langle\text{meaning}\rangle(\langle\text{meaning}\rangle) & \text{(application)} \\
 & \mid & \lambda\langle\text{meaning-var}\rangle.\ \langle\text{meaning}\rangle & \text{(abstraction)} \\
 & \mid & \langle\text{meaning}\rangle \times \langle\text{meaning}\rangle & \text{(product)} \\
\langle\text{type}\rangle & ::= & \langle\text{e-term}\rangle\mid\langle\text{t-term}\rangle\mid\langle\text{t-var}\rangle & \text{(atomic types)} \\
 & \mid & \langle\text{type}\rangle \multimap \langle\text{type}\rangle & \text{(linear implication)} \\
 & \mid & \langle\text{type}\rangle \otimes \langle\text{type}\rangle & \text{(multiplicative conjunction)} \\
 & \mid & \forall\langle\text{t-var}\rangle_1.\langle\text{type}\rangle & \text{(universal quantification over terms from } \langle\text{type}\rangle\text{)} \\
\langle\text{glue}\rangle & ::= & \langle\text{meaning}\rangle\text{: } \langle\text{type}\rangle &
\end{array}
$$

A.2. *Proof Rules for* MILL

	Elimination	Introduction
Implication ($\multimap$)	$\dfrac{\begin{matrix}\vdots \\ A\end{matrix} \qquad \begin{matrix}\vdots \\ A \multimap B\end{matrix}}{B}\multimap_{\mathcal{E}}$	$\dfrac{\begin{matrix}[A]^1 \\ \vdots \\ B\end{matrix}}{A \multimap B}\multimap_{\mathcal{I},1}$
Conjunction ($\otimes$)	$\dfrac{\begin{matrix}\vdots \\ A \otimes B\end{matrix} \qquad \begin{matrix}[A]^1 \ [B]^2 \\ \vdots \\ C\end{matrix}}{C}\otimes_{\mathcal{E},1,2}$	$\dfrac{\begin{matrix}\vdots \\ A\end{matrix} \qquad \begin{matrix}\vdots \\ B\end{matrix}}{A \otimes B}\otimes_{\mathcal{I}}$
Universal ($\forall$)	$\dfrac{\begin{matrix}\vdots \\ \forall x.A\end{matrix}}{A[c/x]}\forall_{\mathcal{E}}$ c free for x	$\dfrac{\begin{matrix}\vdots \\ A[y/x]\end{matrix}}{\forall x.A}\forall_{\mathcal{I}}$ y free for x, y not free in A

A.3. *Meaning Language Term Assignments for* MILL

	Elimination	Introduction
Implication ($\multimap$)	$\dfrac{\begin{matrix}\vdots \\ a : A\end{matrix} \qquad \begin{matrix}\vdots \\ f : A \multimap B\end{matrix}}{f(a) : B}\multimap_{\mathcal{E}}$	$\dfrac{\begin{matrix}[x : A]^1 \\ \vdots \\ f : B\end{matrix}}{\lambda x.f : A \multimap B}\multimap_{\mathcal{I},1}$
Conjunction ($\otimes$)	$\dfrac{\begin{matrix}\vdots \\ a : A \otimes B\end{matrix} \qquad \begin{matrix}[x : A]^1 \ [y : B]^2 \\ \vdots \\ f : C\end{matrix}}{\mathsf{let}\ a\ \mathsf{be}\ x \times y\ \mathsf{in}\ f : C}\otimes_{\mathcal{E},1,2}$	$\dfrac{\begin{matrix}\vdots \\ a : A\end{matrix} \qquad \begin{matrix}\vdots \\ b : B\end{matrix}}{a \times b : A \otimes B}\otimes_{\mathcal{I}}$
Universal ($\forall$)	$\dfrac{\begin{matrix}\vdots \\ t : \forall x.A\end{matrix}}{t : A[c/x]}\forall_{\mathcal{E}}$ c free for x	$\dfrac{\begin{matrix}\vdots \\ t : A[y/x]\end{matrix}}{t : \forall x.A}\forall_{\mathcal{I}}$ y free for x, y not free in A

References

Abney, S. P.: 1987, *The English Noun Phrase in Its Sentential Aspect*., Ph.D. thesis, MIT, Cambridge, MA.

Abramsky, S.: 1993, 'Computational Interpretations of Linear Logic', *Theoretical Computer Science* **111**, 3–57.

Asudeh, A.: 2004, 'Resumption as Resource Management', Ph.D. thesis, Stanford University, CA.

Asudeh, A., and R. Crouch: 2002a, 'Coordination and Parallelism in Glue Semantics: Integrating Discourse Cohesion and the Element Constraint', in M. Butt and T. H. King (eds.), *Proceedings of the LFG02 Conference*, pp. 19–39, CSLI Publications, Stanford, CA.

Asudeh, A., and R. Crouch: 2002b, 'Derivational Parallelism and Ellipsis Parallelism', in Mikkelsen and Potts (eds.), pp. 1–14.

Asudeh, A., and R. Crouch: 2002c, 'Glue Semantics for HPSG', in F. van Eynde, L. Hellan, and D. Beermann (eds.), *Proceedings of the 8th International HPSG Conference*, CSLI Publications, Stanford, CA.

Asudeh, A., and F. Keller: 2001, 'Experimental Evidence for a Predication-Based Binding Theory', in M. Andronis, C. Ball, H. Elston, and S. Neuvel (eds.), *CLS 37: The Main Session*, Vol. 1, pp. 1–14, Chicago Linguistic Society, Chicago, IL.

Barker, C.: 1991, *Possessive Descriptions*, Ph.D. thesis, University of California, Santa Cruz.

Barker, C.: 1995, *Possessive Descriptions*, CSLI Publications, Stanford, CA. Revised version of 1991 UC Santa Cruz, Ph.D. dissertation.

Barker, C., and D. Dowty: 1993, 'Non-Verbal Thematic Proto-Roles', in A. J. Schafer (ed.), *Proceedings of NELS 23*, pp. 49–62, GSLA, Amherst, MA.

Barss, A.: 1986, *Chains and Anaphoric Dependence*, Ph.D. thesis, MIT, Cambridge, MA.

Benton, N., G. Bierman, V. de Paiva, and M. Hyland: 1993, 'A Term Calculus for Intuitionistic Linear Logic', in *Proceedings of the First International Conference on Typed Lambda Calculus*, Vol. 664 of *Lecture Notes in Computer Science*, Springer Verlag, Berlin.

Bresnan, J.: 1984, 'Bound Anaphora on Functional Structures'. Presented at the Tenth Annual Meeting of the Berkeley Linguistics Society.

Bresnan, J.: 1995, 'Linear Order, Syntactic Rank, and Empty Categories: On Weak Crossover', in Dalrymple et al. (eds.), pp. 241–274.

Bresnan, J.: 2001, *Lexical-Functional Syntax*, Blackwell, Oxford.

Butt, M., and T. H. King (eds.): 2001, *Proceedings of the LFG01 Conference*, CSLI Publications, Stanford, CA.

Carpenter, B.: 1997, *Type-Logical Semantics*, MIT Press, Cambridge, MA.

Chao, W., and P. Sells: 1983, 'On the Interpretation of Resumptive Pronouns', in: P. Sells and C. Jones (eds.), *The Proceedings of NELS 13*, pp. 47–61, GLSA, Amherst, MA.

Chomsky, N.: 1993, 'A Minimalist Program for Linguistic Theory', in K. Hale and S. J. Keyser (eds.), *The View from Building 20*, pp. 1–52. MIT Press, Cambridge, MA, Reprinted as chapter 3 of Chomsky (1995).

Chomsky, N.: 1995, *The Minimalist Program*, MIT Press, Cambridge, MA.

Cooper, R.: 1979, 'The Interpretation of Pronouns', in F. Heny and H. Schnelle (eds.), *Selections from the Third Groningen Round Table*, Vol. 10 of *Syntax and Semantics*, pp. 61–92, Academic Press, New York.

Crouch, R., and J. van Genabith: 1999, 'Context Change, Underspecification, and the Structure of Glue Language Derivations', in Dalrymple (ed.), pp. 117–189.

Crouch, R., and J. van Genabith: 2000, 'Linear Logic for Linguists'. Ms., PARC and Dublin City University. http://www2.parc.com/istl/members/crouch/; checked 10/02/2005.

Curry, H. B., and R. Feys: 1958, *Combinatory Logic*, Vol. 1. Amsterdam, North-Holland.

Curry, H. B., and R. Feys: 1995, 'The Basic Theory of Functionality. Analogies with Propositional Algebra', in de Groote (ed.), pp. 9–13. Reprint of Curry and Feys (1958, Chapter 9, Section E).

Dalrymple, M.: 1993, *The Syntax of Anaphoric Binding*, CSLI Publications, Stanford, CA.

Dalrymple, M. (ed.): 1999, *Semantics and Syntax in Lexical Functional Grammar: The Resource Logic Approach*, MIT Press, Cambridge, MA.

Dalrymple, M.: 2001, *Lexical Functional Grammar*, Academic Press, San Diego, CA.

Dalrymple, M., V. Gupta, J. Lamping, and V. Saraswat: 1999a, 'Relating Resource-Based Semantics to Categorial Semantics', in Dalrymple (ed.), pp. 261–280.

Dalrymple, M., R. M. Kaplan, and T. H. King: 2001, 'Weak Crossover and the Absence of Traces', in Butt and King (ed.), pp. 66–82.

Dalrymple, M., R. M. Kaplan, J. T. Maxwell, and A. Zaenen (eds.): 1995, *Formal issues in Lexical-Functional Grammar*, CSLI Publications, Stanford, CA.

Dalrymple, M., J. Lamping, F. Pereira, and V. Saraswat: 1999b, 'Overview and Introduction', in Dalrymple (ed.), pp. 1–38.

Dalrymple, M., J. Lamping, F. Pereira, and V. Saraswat: 1999c, 'Quantification, Anaphora, and Intensionality', in Dalrymple (ed.), pp. 39–89.

Dalrymple, M., J. Lamping, and V. Saraswat: 1993, 'LFG Semantics via Constraints', in *Proceedings of the Sixth Meeting of the European ACL*, University of Utrecht, pp. 97–105.

de Groote, P. (ed.): 1995, *The Curry–Howard Isomorphism*, Vol. 8 of *Cahiers du Centre de Logique*, Academia, Louvain-la-neuve, Belgium.

Elbourne, P.: 2001, 'E-Type Anaphora as NP-deletion', *Natural Language Semantics* **9**, 241–288.

Elbourne, P.: 2002, *Situations and Individuals*, Ph.D. thesis, MIT, Cambridge, MA.

Engdahl, E.: 1985, 'Parasitic Gaps, Resumptive Pronouns, and Subject Extractions', *Linguistics* **23**, 3–44.

Evans, G.: 1977, 'Pronouns, Quantifiers, and Relative Clauses (I)'. *Canadian Journal of Philosophy* **7**(3), 467–536. Reprinted in Evans (1985, pp. 76–152).

Evans, G.: 1985, *Collected Papers*, Clarendon Press, Oxford. Antonia Phillips (ed.).

Fox, D., and J. Nissenbaum: 2004, 'Condition A and Scope Reconstruction', *Linguistic Inquiry* **35**(3), 475–485.

Frank, A., and J. van Genabith: 2001, 'LL-based Semantics Construction for LTAG – and what it teaches us about the relation between LFG and LTAG', in Butt and King (eds.), pp. 104–126.

Gallier, J.: 1995, 'On the Correspondence between Proofs and λ-terms', in de Groote (ed.), pp. 55–138.

Girard, J.-Y.: 1987, 'Linear Logic', *Theoretical Computer Science* **50**, 1–102.
Girard, J.-Y.: 1995, 'Linear Logic: A Survey', in de Groote (ed.), pp. 193–255.
Halvorsen, P.-K., and R. M. Kaplan: 1988, 'Projections and Semantic Description in Lexical-Functional Grammar', in *Proceedings of the International Conference on Fifth Generation Computer Systems*, Tokyo, pp. 1116–1122.
Heim, I., and A. Kratzer: 1998, *Semantics in Generative Grammar*, Blackwell, Oxford.
Hepple, M.: 1990, 'The Grammar and Processing of Order and Dependency: A Categorial Approach'. Ph.D. thesis, University of Edinburgh, UK.
Higginbotham, J.: 1984, 'English is not a Context-Free Language'. *Linguistic Inquiry* **15**(2), 225–234.
Hindley, J. R., and J. P. Seldin: 1986, *Introduction to Combinators and λ-Calculus*, Vol. 1 of *London Mathematical Society Student Texts*, Cambridge University Press, Cambridge.
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, Academic Press, London. Circulated in unpublished form from 1969. Reprinted in de Groote (1995, pp. 15–26).
Jacobson, P.: 1999, 'Towards a Variable-Free Semantics', *Linguistics and Philosophy* **22**, 117–184.
Jacobson, P.: 2002, 'The (Dis)organization of the Grammar: 25 Years', *Linguistics and Philosophy* **25**, 601–626.
Jacobson, P.: 2003, 'Direct Compositionality and Variable-Free Semantics: The Case of "Principle B"'. Ms., Brown University. Presented at the Workshop on Direct Compositionality, Brown University, Providence, Rhode Island, June 2003.
Janssen, T. M. V.: 1997, 'Compositionality', in van Benthem and ter Meulen (1997), pp. 417–473. Co-published with Elsevier Science B.V., Amsterdam, The Netherlands.
Jensen, P. A., and C. Vikner: 1994, 'Lexical Knowledge and the Semantic Analysis of Danish Genitive Constructions', in S. L. Hansen and H. Wegener (eds.), *Topics in Knowledge-based NLP-systems*, pp. 37–55, Copenhagen, Samfundslitteratur.
Kameyama, M.: 1985, *Zero Anaphora: The Case of Japanese*, Ph.D. thesis, Stanford University, CA.
Kaplan, R. 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, Academic Press, London, Reprinted in Dalrymple et al. (1995, pp. 339–367).
Kaplan, R. M.: 1989, 'The Formal Architecture of Lexical-Functional Grammar', in C.-R. Huang and K.-J. Chen (eds.), *Proceedings of ROCLING II*, pp. 3–18. Reprinted in Dalrymple et al. (1995, pp. 7–27).
Kaplan, R. M., and J. Bresnan: 1982, 'Lexical-Functional Grammar: A Formal System for Grammatical Representation', in J. Bresnan (ed.), *The Mental Representation of Grammatical Relations*, pp. 173–281, MIT Press, Cambridge, MA. Reprinted in Dalrymple et al. (1995, pp. 29–130).
Kaplan, R. M., and J. T. Maxwell: 1996, 'LFG Grammar Writer's Workbench', Technical report, PARC, Palo Alto, CA. ftp://ftp.parc.xerox.com/pub/lfg/lfgmanual.ps; checked 10/02/2005.
Keenan, E. L., and B. Comrie: 1977, 'Noun Phrase Accessibility and Universal Grammar', *Linguistic Inquiry* **8**, 63–99.

Keller, F., and A. Asudeh: 2001, 'Constraints on Linguistic Coreference: Structural vs. Pragmatic Factors', in *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, Lawrence Erlbaum, Mahwah, NJ.

Klein, E., and I. A. Sag: 1985, 'Type-Driven Translation'. *Linguistics and Philosophy* **8**, 163–201.

Kokkonidis, M.: 2003, 'Glue and λDRT for Meaning Assembly', Ms., Department of Philosophy, King's College, London.

Kokkonidis, M.: 2004, 'First-Order Glue', Ms., Cambridge University, appear in *Journal Logic, Language and Information*.

Lebeaux, D.: 1988, *Language Acquisition and the Form of the Grammar*, Ph.D. thesis, University of Massachusetts, Amherst.

Martí, L.: 2001, *Contextual Variables*, Ph.D. thesis, University of Connecticut.

Martí, L.: 2003, *Contextual Variables as Pronouns*, in R. B. Young and Y. Zhou (eds.), *Proceedings of SALT 13*, Cornell Linguistics Circle Publications, Ithaca, NY.

Martí, L.: 2004, 'The Syntactic Presence of Contextual Variables'. Ms., University of the Witwatersrand, Johannesburg.

McCloskey, J.: 1979, *Transformational Syntax and Model Theoretic Semantics: A Case-Study in Modern Irish*, Dordrecht: Reidel.

McCloskey, J.: 1990, 'Resumptive Pronouns, $\overline{\text{A}}$-Binding and Levels of Representation in Irish', in R. Hendrick (ed.): *Syntax of the Modern Celtic languages*, Vol. 23 of *Syntax and Semantics*, pp. 199–248, Academic Press, San Diego, CA.

McCloskey, J.: 2002, 'Resumption, Successive Cyclicity, and the Locality of Operations', in S. D. Epstein and T. D. Seeley (eds.), *Derivation and Explanation in the Minimalist Program*, pp. 184–226, Blackwell, Oxford.

Mikkelsen, L., and C. Potts (eds.), 2002, 'Proceedings of the 21st West Coast Conference on Formal Linguistics'. Cascadilla Press, Somerville, MA.

Mitchell, J.: 1986, *The Formal Semantics of Point of View*, Ph.D. thesis, University of Massachusetts, Amherst.

Montague, R.: 1974, 'The Proper Treatment of Quantification in Ordinary English', in *Formal Philosophy: Selected Papers of Richard Montague*, Yale University Press, New Haven, CT. Richmond H. Thomason (Ed.),

Moortgat, M.: 1997, 'Categorial Type Logics', in van Benthem and ter Meulen (1997), pp. 93–177, Co-published with Elsevier Science B.V., Amsterdam, The Netherlands.

Morrill, G.: 1994, *Type Logical Grammar*, Kluwer, Dordrecht.

Partee, B. H.: 1983/1997, 'Genitives – A Case Study', in van Benthem and ter Meulen (1997), pp. 464–470. Appendix to Janssen (1997). Published version of 1983 manuscript.

Partee, B. H.: 1989, 'Binding Implicit Variables in Quantified Contexts', in C. Wiltshire, B. Music, and R. Graczyk (eds.), *Papers from CLS 25*, pp. 342–365, Chicago Linguistic Society, Chicago, IL.

Partee, B. H., and V. Borschev: 1998a, 'Integrating Lexical and Formal Semantics'. Class lectures, International Ph.D. Course on Integrating Lexical and Formal Semantics. Kolding, Denmark, Southern Denmark Business School.

Partee, B. H., and V. Borschev: 1998b, 'Integrating Lexical and Formal Semantics: Genitives, Relational Nouns, and Type-shifting', in *Proceedings of the Second Tbilisi Symposium on Language, Logic, and Computation*, Tbilisi, pp. 229–241.

Partee, B. H., and V. Borschev: 2003, 'Genitives, Relational Nouns, and Argument-Modifier Ambiguity', in E. Lang, C. Maienborn, and C. Fabricius-Hansen (eds.), *Modifying Adjuncts*, Vol. 4 of *Interface Explorations*, Mouton de Gruyter, Berlin.

Pullum, G. K.: 1985, 'Such that Clauses and the Context-Freeness of English'. *Linguistic Inquiry* **16**(2), 291–298.

Runner, J. T., R. S. Sussman, and M. K. Tanenhaus: 2002, 'Logophors in Possessed Picture Noun Phrases', in Mikkelsen and Potts (eds.), pp. 401–414.

Runner, J. T., R. S. Sussman, and M. K. Tanenhaus: 2003, 'Assignment of Reference to Reflexives and Pronouns in Picture Noun Phrases: Evidence from Eye Movements'. *Cognition* **89**, B1–B13.

Sells, P.: 1984, *Syntax and Semantics of Resumptive Pronouns*, Ph.D. thesis, University of Massachusetts, Amherst.

Shlonsky, U.: 1992, 'Resumptive Pronouns as a Last Resort', *Linguistic Inquiry* **23**, 443–468.

Stanley, J.: 2000, 'Context and Logical Form', *Linguistics and Philosophy* **23**(4), 391–434.

Stockwell, R., P. Schachter, and B. H. Partee: 1973, *The Major Syntactic Structures of English*, Holt, Rinehart and Winston, NY.

Storto, G.: 2003, *Possessives in Context: Issues in the Semantics of Possessive Constructions*, Ph.D. thesis, University of California, Los Angeles.

Troelstra, A. S.: 1992, *Lectures on Linear Logic*, CSLI Publications, Stanford, CA.

van Benthem, J., and A. Meulen (eds.): 1997, *Handbook of Logic and Language*, MIT Press, Cambridge, MA. Co-published with Elsevier Science B.V., Amsterdam, The Netherlands.

van Genabith, J., and R. Crouch: 1999, 'Dynamic and Underspecified Semantics for LFG', in Dalrymple (ed.), pp. 209–260.

Vikner, C., and P. A. Jensen: 2002, 'A Semantic Analysis of the English Genitive. Interaction of Lexical and Formal Semantics', *Studia Linguistica* **56**, 191–226.

Willis, D.: 2000, 'On the Distribution of Resumptive Pronouns and Wh-trace in Welsh', *Journal of Linguistics* **36**, 531–573.

Institute of Cognitive Science
2201 Dunton Tower
Carleton University
Ottawa, ON K1S 5B6
Canada
E-mail: asudeh@csli.stanford.edu